FEBRUARY/MARCH 1984 An independent magazine published by ECC Publications

95p

SINCLAIR PROJECTS

THE MAGAZINE FOR THE SERIOUS SINCLAIR USER

CHARACTER GENERATOR
TAPE HEAD RENEWAL
ZX-81 SOUND BOARD
AUTOMATIC RESTART
RAM RE-SET

CATCHING THE WIND— DIRECTION RECORDER

EXCLUSIVE OFFER— ZX-81 KIT FOR ONLY £25

SINCLAIR PROJECTS

FROM THE EDITOR

DURING the warm summer months you may wish to be reminded of the poor weather at the beginning of the year. Graham Bradley, in the second article in his series on building a weather station controlled by a Spectrum, shows how to make a wind direction indicator.

The device allows you to record the differing directions of the wind and the length of time it has been blowing from a particular direction. This month we concentrate on the hardware part of the project and in future issues we will be publishing software systems which will allow you to make use of the information which has been collected.

Bradley has also returned to another of our previous issues for a project. Last month we showed how to make a sound generator to improve the sound capabilities for the Spectrum. That has now been adapted for the ZX-81. We include two sample programs to show how good use can be made of this peripheral.

Two of this month's projects allow programs to be stored in RAM after the power has been cut off either deliberately or accidentally. Raymond Hopkins' device is important for anyone wishing to use a ZX-81 as a permanent control system, when an interruption in power would cause it to crash with consequences which could be disastrous. As soon as power is restored, the peripheral ensures that the system starts up again automatically.

Bradley has taken the battery-backed RAM from the previous issue and enhanced it by adding a device which, when switched-on, runs a machine code program by pressing a single key.

Our final two projects are simple articles which help to improve your ability to use your machines. Charles Barnatt deals with a subject with which many people have difficulty — the SAVEing and LOADing of tapes. One of the reasons for many problems is the poor state of cassette recorder heads. We show how to replace them to give your recorder a longer life.

Mike Biddell has written a software project for improving the letters on the screen and printouts.

Sinclaur Projects is now more than one year old and during our first 12 months we have learned a great deal about what people are doing with their Spectrums and ZX-81 and have been pleasantly surprised at the wide variety of uses which are being made of what many people consider to be only toys.

Editor Nigel Clark **Consultant editor** David Buckley **Production editor** Harold Mayes MBE **News writer** Stephen Adams **Design** Elaine Bishop **Cover** Stuart Briers **Advertisement manager** John Ross **Advertisement Executive** Annette Burrows **Editorial assistant** Dezi Epaminondu **Managing director** Terry Cartwright **Chairman** Richard Hease.

Sinclair Projects is published bi-monthly by ECC Publications Ltd. It is in no way connected with Sinclair Research Ltd.

Telephone, all departments: 01-359 3525. If you would like to contribute to any of the *Sinclair User* group of publications please send programs, articles or ideas for hardware projects to Sinclair User and Projects, ECC Publications, 196–200 Balls Pond Road, London N1 4AQ. We pay £50 per 1,000 words for each article used.

ISSN 0264/0449. Printed and typeset by Cradley Print PLC, Warley, W. Midlands. Distributed by Spotlight Magazine Distribution Ltd, 1 Benwell Road, Holloway, London N7. 01-607 6411.

AGF

MICRODRIVE
O.K. FOR ALL IS

PROGRAMMABLE JOYSTICK INTERFACE for Spectrum or ZX81

ONLY 32·95 +£1·00pp

NOW AVAILABLE from SELECTED RETAILERS

PROGRAMMABLE INTERFACE

The AGF Programmable Joystick Interface is a unique design offering the use of any Atari-compatible joystick with absolutely all software, whether it is cassette or ROM cartridge, with the Sinclair Spectrum or ZX81.

The hardware programmable interface requires no additional software and accurately replicates the keys of the computer in a manner which is responsive to absolutely *ALL* key reading methods, both BASIC and Machine Code.

The interface does not interfere with key operation and can therefore be used simultaneously with the keyboard.

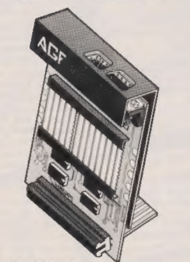
There is no need to remove the interface once fitted as the rear extension connector will accommodate further expansion, i.e printers or RAM packs etc. This important feature avoids excessive wear to the expansion port.

The key replication principle pioneered by AGF means that your own programs can use eight directional joystick movement by utilising simple key reading BASIC.

Two joystick sockets are provided which share the same keys, for use with the majority of two player games. Several interfaces may be used on the same computer for multiple joystick applications.

The interface is programmed by a two digit code, which is looked up on a programming chart supplied, for each direction and firing button. The two numbers are then selected on a pair of leads which are clipped onto appropriately numbered strips on the interface.

Once configured this can be marked on a Quick Reference Programming Card for storing with the game. As the programming is *not* power dependent the interface retains the last configuration made and can be immediately used when next switched on.



KEY FEATURES

* Programmable design gives TOTAL software support.
* Accepts Atari, Competition Pro, Wico, Starfighter, Quick Shot, Le Stick etc.
* Rear extension connector for all other add-ons.
* Free demo program and instructions.

PACKAGE CONTENTS SUPPLIED

- Programmable Interface Module as illustrated, complete with clip-on programming leads.
- Self adhesive programming chart detailing how to define which key is simulated by UP, DOWN, LEFT, RIGHT, and FIRE. This can be fixed on to the case of your computer or if preferred the protective backing can be left on. The chart is made of a very durable reverse printed plastic and is extremely easy to read.
- One pack of ten Quick Reference Programming Cards for at-a-glance setting to your games requirements. The card allows you to mark the configuration in an easy to read fashion with space to record the software title and company name.
- Video Graffiti demonstration program which is written totally in BASIC to illustrate how all eight directions and fire can be read. This is also a useful high resolution drawing program.
- 12 months guarantee and full written instructions.

JOYSTICKS CONTROLLERS

FOR USE WITH OUR INTERFACE Module or VIC 20, Commodore 64, Atari VCS, Atari 400, Atari 800

If you require extra Joysticks for our original interface module mark order 'OLD' Joysticks

ONLY £7.54 inc VAT + P&P

PLEASE ALLOW 28 DAYS FOR DELIVERY — **ALL ORDERS CONFIRMED**

FROM: MR/MRS/MISS

ADDRESS

SEND C.W.O. (NO STAMP NEEDED) TO: A.G.F. HARDWARE, DEPT. SU

FREEPOST, BOGNOR REGIS, WEST SUSSEX, PO22 9BR

QTY	ITEM	ITEM PRICE	TOTAL
	PROGRAMMABLE INTERFACE	33.95	
	JOYSTICK(S)	7.54	
	PACK(S) QUICK REFERENCE CARDS	1.00	
ONE	VIDEO GRAFFITI	FREE	
	ZX81 ☐ ZX SPECTRUM ☐ Please tick	FINAL TOTAL	

DEALER ENQUIRIES WELCOME *EXPORT PRICES ON APPLICATION*

COMPATIBILITY

SUE SPECTRUMS

Computer Trade Association Member

JOYSTICK INTERFACE II for Spectrum or ZX81

NEW PRICE

ONLY ~~19.95~~ +£1.00pP

15.95

JOYSTICK INTERFACE

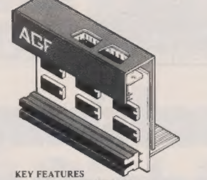
The Interface Module II has been specially designed to plug on to the rear connector of your ZX Spectrum or ZX81 and allow you to connect any standard Atari type digital Joysticks. All of the computer's connections are duplicated on an extension connector so that you can still use any other devices intended for use with your computer.

The Interface Module II resides in the same memory space as the keyboard, which remains fully functional at all times, therefore it will not interfere with anything else connected.

When a suitable joystick is plugged into 'Player 1' socket its action will mimic pressing the cursor keys, up "7", left "5" and so on. The firing button will simulate key 0. This unique feature guarantees the best software support.

Take a look at the selection of compatible games we have listed. More are being added all the time as a result of our contact with the various software companies.

A second Joystick may be connected in the 'Player 2' position which simulates in a parallel fashion keys T-Y-U-I-P. This will allow you to play a whole new generation of two player games.



KEY FEATURES

* Proven cursor key simulation for maximum software support
* Accepts Atari, Competition Pro, Wico, Starfighter, Le Stick, etc Joysticks
* Second Joystick facility
* Rear extension connector for all other add-ons

AGF COMPATIBLE SOFTWARE – AVAILABLE NATIONWIDE

The following titles are available from us:

Galactic Jailbreak/ Snake	: Apocolypse Software	£4.95
3D Tanx	: DK 'Tronics	£4.95
Splat !	: Incentive Software Ltd	£5.50
Pheenix	: Megadodo Software	£5.50
Escape	: New Generation Software	£4.95
3D Tunnel	: " "	£5.95
Knot in 3D	: " "	£5.95
Cyber Rats	: Silversoft	£5.95

COMPATIBILITY CASSETTES £4.95

These cassettes have short programs to load before the chosen game which will convert it to use the cursor keys and therefore become compatible with the Interface Module II.

Cassette 1 converts	Cassette 2 converts
Arcadia	Centipede
Schizoids	Planetoids
Hungry Horace	Jet-Pac
Horace Goes Skiing	† Pssst
Spectres	† 3D Combat Zone
Penetrator	† Invaders

† *Will require 48K Memory.*

FROM: MR/MRS/MISS

ADDRESS

SEND C.W.O. (NO STAMP NEEDED) TO: A.G.F. HARDWARE, DEPT. SU

FREEPOST, BOGNOR REGIS, WEST SUSSEX, PO22 9BR

QTY	ITEM	ITEM PRICE	TOTAL
	INTERFACE MODULE II	16.95	
	JOYSTICK(S)	7.54	
	SOFTWARE:		
	SOFTWARE:		
ZX81 ☐	ZX SPECTRUM ☐ Please tick	FINAL TOTAL	

DEALER ENQUIRIES WELCOME *EXPORT PRICES ON APPLICATION*

WHERE TO BUY AGF PRODUCTS OVER THE COUNTER

Ashby Computer Centre
186 Ashby High Street, Scunthorpe, S. Humberside DN16 2JR

Brainwave Microcomputers
24 Crown Street, Ipswich, Suffolk IP1 3LD

Buffer Micro Ltd
310 Streatham High Road, London SW16

Chelsea Micros Ltd
14 Jerdan Place, London SW6 1BH

Computers of Wigmore Street
87 Wigmore Street, London W1H 9FA

Everybodys Hobbies
1 Great Colman Street, Ipswich, Suffolk IP4 2AA

4Mat Computing
67 Friargate, Preston, Lancashire PR1 2AT

Gamer
24 Gloucester Road, Brighton BN1 4AQ

GB Microland
7 Queens Parade, London Road, Waterlooville, Hants

Melgray Hi-Tech Ltd
49 Broad Street, Hereford HR4 9AR

Micro Fare
296 Gloucester Road, Horfield, Bristol

Raven Video
74 Green Lane, Tettenhall, Wolverhampton

Screen Scene
144 St George's Road, Cheltenham Gloucestershire GL50 3EL

Screens
8 Main Avenue, Moor Park, Northwood Middlesex.

Syntax Computers
76 Cornwall Street, Plymouth PL1 1NS

Teleco Video
53 Maple Road, Penge, London SE20

Telford Electronics & Computing
26a Bradford Street, Shipnal, Shropshire TF11 8AU

The Computer Shop
Unit 25, Handyside Arcade, Percy Street, Newcastle-upon-Tyne NE1 4PZ

The Computer Centre (Humberside) Ltd
26 Anlaby Road, Hull HU1 2PA

GILSOFT | COMPUTASOLVE LTD. | KEYSOFT

cp software | HEWSON CONSULTANTS

INTERFACE PUBLICATIONS | SUNSHINE | SOFTEACH | Hilderbay

EAST LONDON ROBOTICS | COMPUSOUND | Melbourne

CDS Micro Systems | Heinemann Educational Books | LCL | QS THE GAME LORDS | CAMBRIDGE COMPUTING | dk'tronics

Elephant Software. | FOX ELECTRONICS | AGF | FAL-SOFT COMPUTERS

Look at some of the
if you don't visit the

Are you missing out on everything that's right for your ZX machine?
Do you want to see the latest developments in hardware, software, books, peripherals and gizmos?
Then there's only one venue that's a must for you.
The 10th ZX MICROFAIR at ALLY PALLY on the 4th February 1984!

GET YOUR ADVANCE TICKETS NOW!

10th ZX Microfair.
Ally Pally February 4th.

Saturday 4th February 10am–6pm
ALEXANDRA PAVILION, ALEXANDRA PARK, WOOD GREEN, LONDON N22

Advance Tickets from: Mike Johnston, 71 Park Lane, Tottenham, London N17 0HG.
Prices: Adults £1 Kids (under 14) 50p. Parties of 10 or more at 10% discount!
Please make cheques payable to ZX MICROFAIR and enclose S.A.E.

Prices at the door: Adults £1.25, Kids 60p

Exhibitors — phone Mike Johnston on 01-801 9172 for details!

Second Edition Just Published

If you only buy ONE computing book

THIS IS THE CLASSIC!

An entertaining, but **thorough** reference source with the most readable, comprehensive glossary you'll find anywhere. The Good Computing Book for Beginners is an essential A-Z of all the facts you need to know about computing — and none you don't!

Whether you're brand new to computing, or an old hand keen to stay ahead, you'll score by keeping this classic, top-selling book within reach to use again and again.

The author, Dennis Jarrett, is a successful journalist who was also founder editor of Which Computer? magazine — so his first edition quickly became a standard work. Now this new, substantially enlarged and revised edition covers the latest trends, terms and technology with the relevant facts — and ONLY the relevant facts — in **plain English.**

Here's another fact you'll find fascinating: it will only cost you £2.95!

Use the coupon below to get your copy right away — or buy it soon from your bookshop. It's the **one** book you'll turn to again and again.

Please send me ______ copy(ies) of The Good Computing Book for Beginners by Dennis Jarrett at £2.95 each plus 50p post and packing. I enclose a cheque for ______ / Please debit my credit card account Visa/Access/Diners/Amex number ______

Signature ______

Name ______

Address ______

______ (please print clearly)

Send to: ECC PUBLICATIONS LIMITED
196–200 Balls Pond Road
London N1 4AQ

Interface 2 lacks thought

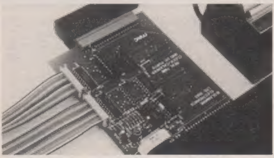
SINCLAIR has announced the Interface Two, which followed the Interface One within a few weeks. Interface Two provides two joystick sockets which take standard Atari joysticks and a ROM cartridge slot. The joystick ports are non-standard, as the only software to work with them is from Sinclair or Psion.

They operate the number key only and each joystick operates the first or last set of numbers. Despite the advertisements there is no software built into it as Interface Two is only a ULA to act as a port for the joysticks. The ROM cartridge socket is also a disappointment, as it provides no special switching to page ROMs in and out as happens on the BBC micro.

The cartridge has all 16 address lines and eight data lines on it but because of the way Sinclair designed the Spectrum, none of the internal memory can be switched off, so the only programs which can be provided can be 16K-long versions which will replace the Basic ROM.

They are available by turning-off the power to the Spectrum, plugging-in the cartridge and turning-on the power. The games then autostart — only three were available at the time of the launch, with another set available the following week. The cartridges available were Space Raiders (invaders), Backgammon and Planetoids (asteroids). All would become uninteresting after a few attempts.

At £19.95 for the Interface Two and £14.95 for each ROM cartridge, they may not be in great demand unless the price is reduced dramatically. The joysticks are also out of step with the current trend of software manufacturers who program their games to work with the Kempston-type joystick interface.



Counting the cost of making use of Glanmire's time

GLANMIRE Electronics of the Republic of Ireland has produced a real-time clock and eight-bit I/O port for the Spectrum or ZX-81. The crystal-controlled clock is battery-driven and is recharged when connected to the computer power supply — a maximum of 40ma according to the booklet.

It is set to compensate for various months but not leap years. It can be adjusted by a screwdriver control on the board to go faster or slower. A 256-byte PROM mounted on the board provides the software to read and write to the clock from within Basic and the time is returned in a Basic variable called T$.

That contains month number — characters 1 and 2; day number — next three characters; date — next two; hour — next two; minutes — next two; and seconds — last two. Print T$ (x TO y) will produce the parts of the clock required for the program.

Each access to the clock is via a call to a USR routine which loads T$ with the time. T$ will not be changed until the next USR call. T$ is also used to set the time by writing it into the clock via another user call.

Any writing to the clock can be prevented by omitting the write peg on the board. That should also be disconnected when powering the computer up and down. It has no effect on reading the clock and also NEW has no effect on the clock routine.

The board also has a single eight-bit read-and-write I/O port which seems to cover all the address lines A7-A5 not used by the clock. They appear as 16 Molex-type connector pins on the side of the board. Examples of programs using the clock and the I/O port are given in great detail but only three projects which the user can build are listed.

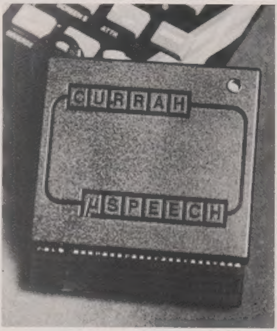
They are a moving-hand clock, a stopwatch using the port and an intruder alarm. Various other circuit diagrams for sensors and switches are given but no real explanation of how they work.

It provides a good basis for experimenting but at a cost. It is also incompatible with most other I/O equipment for the Spectrum, as it uses all the spare addresses.

From Glanmire Electronics Ltd, Meenane, Watergrasshill, Co. Cork, Ireland, the clock and I/O ports costs £40 for the Spectrum version and for the ZX-81 version £36. They are not interchangeable.

NEWS

Learning to speak currah



THE CURRAH U Speech Module is a black plastic box which plugs into the back of a Spectrum and gives an amazing range of facilities. The unit is approximately 3in. square and ½in. high. Once plugged into the Spectrum expansion port, no more expansion is possible unless a motherboard is used, as it lies flat behind the Spectrum.

There are two leads from the unit. One goes into the aerial socket and the other into the MIC socket. The lead to the TV is plugged into the socket on the back of the unit. That must be done before powering-up the Spectrum.

The unit provides its own copyright message at the top of the screen when powered-up and pressing a key will also start the unit into its keyvoice mode. That is where every key used is spoken by the unit. All the keywords except the tilth ( ) are spoken — even the direction arrows which come out as CURSOR. The colours, of course, are not spoken or the various modes.

The keyvoice is controlled by a variable called KEYS and can be turned-off by LET KEYS = 0. That can be used directly or in the program and the keyvoice can be turned on again by LET KEYS = 1. The keyvoice also works for keys pressed in the INPUT or INKEY$ unless disabled.

Another useful feature is that S$ has been allocated as a speech buffer and the LET S$ command makes S$ into a spoken string. Only letters are allowed, which is a pity, since numbers would be useful. Letters can also be used in brackets to give single or double allophones. An allophone is a sound rather than a letter in speech and words must be programmed to sound correct.

Most words will be satisfactory if typed-in directly but Os, As and some others may need a set of allophones instead. In that respect the booklet with the unit is very good, giving clear examples and a list of suitable alternatives. Unfortunately the variable KEYS does not effect the speaking of S$.

For the technically-minded, the unit contains a ULA which works on a WRITE command from the microprocessor, a ROM containing the keyword speech patterns and SP0256-AL2 speech processor. It also contains a clock for clear speech and an audio modulator to transfer the sound to the TV lead. The sound can be adjusted by using a screwdriver on the screw showing on the top at the right-hand side of the box.

The U-Speech allocates itself the top 256 bytes of memory at switch-on and moves down the USR graphics and RAMTOP. More can be allocated to that buffer by the use of CLEAR. That makes it incompatible with some programs which use that space for machine code. Details of the buffer are given at the back of the book for machine code users.

A cassette containing demonstration programs and a game should accompany the unit but it was not sent with our copy so that all tests were done from the manual.

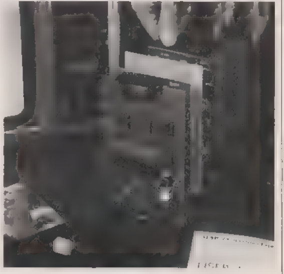
Time must be allowed in all programs for the speech, as it is updated only by the keyboard interrupt routine every 50ms. That also means that during SAVE/LOAD/VERIFY/BEEP and dealings with any device connected to interface one — i.e., Microdrive, RS232 or network — no speech should be in progress. That is because the speech will continue as one sound until the operation is finished.

The unit is extremely useful but time has not permitted it to be tested with any other units to see if they clash. What is presented is a very good clear speech box, with a very easy way of programming it and, even more useful, a spoken response to any key input. That might become a more than essential unit for some disabled users.

The lack of numbers and other characters which are not spoken on output will have to be tolerated.

The Currah U Speech unit costs £29.95 and is inexpensive compared to what it offers. Currah Computer Components is at Greythorp Industrial Estate, Hartlepool, Cleveland TS25 2DF. The company is intending to make it usable for the ZX-81, BBC and other computers.

Ideal growth at high cost



U-MICROCOMPUTERS, an Apple card manufacturer, has decided to do the same thing for the Spectrum. One of the things for which the Apple computer is famous is the internal motherboard-type slots, which allow you to plug-in things like RS232s, parallel ports and disc drives. All are on separate cards, which need only to be plugged into the system. They are also expensive and some have to be put in particular slots.

The same kind of system has been adopted by U-Micro on its Spectrum range. There is a three-slot motherboard which is completely buffered to reduce the load on the Spectrum. An extension is also offered which will extend it up to seven. That, however, must be supplied by a separate power supply using a standard — non-standard to Sinclair users — multi-way plug. The power supply must also supply +12 volts, +9 volts and −12 volts for things like RS232s.

There are, however, advantages. One is that a spare, unbuffered printed circuit board edge on the right-hand side of the board is provided; all edge connectors and cards are gold-plated as a matter of course. That can take the Interface One satisfactorily and would solve a problem for users whose Spectrum is in a case.

The other advantages are complete Spectrum bus compatability, unlike some other systems, and an alternative decoding system for I/O devices which will allow seven boards to be added to one system without clashes.

That is because the motherboard provides for each edge connector, or slot, to have a different chip-select signal on edge connector position 4A.

That is derived from the top three address lines A7-A5. Internal addresses on each card use the upper three address lines A8-A10.

All that, of course, increases the cost but U-Micro also compensates for that by detailing in a glossy booklet all the information you want to know about the card you have bought. The booklet contains not only details on how to set up and use the card but also a circuit diagram and specifications of the major chip used.

Listings are also given of any software included in the package.

For serious work using a Spectrum the board is ideal; it relieves the Spectrum of power supply problems and loading of the edge connector by too many devices. It also allows you more cards than any other systems which are compatible with Sinclair equipment.

The only problem is that the ideal solution is not inexpensive. An adaptor is required to fit the Spectrum and motherboard printed circuit board edges together costs £6.90, three-slot backplane £35.65, four-slot extension — if required — £25.30. A power supply was not available at the time of writing; a £70 alternative was supplied. The Spectrum one should be half the price, U-Micro says. There should be an additional £1.50 for postage and packing.

Cards available from U-Micro are dual RS232, dual parallel ports — with a Centronics kit as an extra — and a prototyping card. Other producers' Spectrum cards can also be used but may restrict the use of addresses.

All equipment has a 12-month guarantee and can be obtained from U-Microcomputers Ltd, Winstanley Industrial Estate, Long Lane, Warrington, Cheshire WA2 8PR. Tel: 0925-54117.

Simple add-on gives new use for power supply

A SIMPLE device is available which allows you to use the ZX-81 or Spectrum +9 volt power supply to power other equipment while you are not using your computer.

Two versions are available, one as a standard +5 volt regulated power supply and the other adjustable from +9 volts to +5 volts. Two screw terminals are provided for the output to your circuits.

It can be used to supply more power to any extra boards on a motherboard system if you have two power supplies. Versions are available for ZX-81 and Spectrum.

The unit is in a small black plastic box $2\frac{3}{4} \times 1\frac{1}{2} \times 1$in. and has an LED on top to indicate power on. The cost is £6.95 for the non-adjustable version and £7.95 for the variable version. Centec Electronic Systems is at 47 Spur Road, Orpington, Kent BR6 0QR. Tel: 0689-35353.

NEWS

Analogue conversion

EPROM SERVICES has produced two digital-to-analogue converters for the ZX-81 and Spectrum. They allow you to program an output voltage up to +9 volts by outputting a number from the computer. One of them is an independent unit which has three A/D converters and requires no extra port to use it. It is a bare board and an edge connector must be soldered on if you are not using a motherboard. The unit is incompatible with the Interface One, which controls the Microdrive, as both use the same address line A4.

The three outputs appear on screw terminals at the top of the board and there are three sets of variable resistors to adjust the zero volts and the maximum volts setting.

The maximum is pre-set to 3.25 volts but can be raised to 9 volts by cutting a track on the printed circuit board and inserting a resistor in the holes provided. A simple calculation, which is detailed in the instructions, is all that is required to find a suitable resistor. All the outputs are buffered and can be short-circuited accidentally but that is not recommended.

The other D/A is very similar but will fit only on to the Mk I version of the company's I/O port. It clips on via its own edge connector to port 'C' and provides one output on the same screw terminals. Adjustment is in the same way via two variable resistors.

There is a disadvantage to the set zero volts resistor on both versions. Both variable resistors are set immediately above one another and so the lower one is accessible only from the underneath, about 2in. away from the edge connector, if using a motherboard.

The three-output D/A board costs £27 and the single output one for use with a Mk I port only £10.

Eprom Services is at 3 Wedgewood Drive, Leeds LS8 1EF. Tel: 0532-667183.

Regulated power for ZX-81

PR ELECTRONICS has produced a small aluminium box which regulates the power supply to a ZX-81 or Spectrum, thus reducing the amount of heat generated inside the computer.

It will help with problems in using the Spectrum with a TV display, as it eliminates most of the dot crawl.

The power pack plugs into the socket on the box and a lead from the box plugs into the computer. All the heat is dissipated through the aluminium box and so a warning label is included.

PR Electronics is at 14 Bretby Close, Bessacar, Doncaster, Yorkshire.

Adaptable variations

TWO MONTHS ago you published an amplification method of which I already knew. I wondered why I did not send the method in first. Now I have furthered my research in amplification and found a way of obtaining a really loud sound from the Spectrum.

It is done with the aid of an ear plug. To amplify, connect the black jack plug to the MIC socket of a Spectrum and the white jack plug to the EAR socket; connect the other white jack plug into the MIC socket of the tape recorder and connect an ear plug to the EAR socket. Then place a blank tape in your recorder, press pause, and set the recorder to record. There is no need to place the ear plug into your ear. I have also found that there is no need to use the MIC sockets when LOADing software.

James Collins, aged 13, Laindon, Essex.

● *You omitted to say what make of tape recorder you use but when it was tried on a very old Sanyo, connecting the EAR or MIC sockets of the Spectrum to the MIC input of the recorder and setting the recorder to PLAY without a tape worked well. It did not work when on RECORD. If you try this be careful of over-loading the MIC input to the tape recorder. The sound will be distorted if that is happening and it could damage the recorder.*

Hi-res board

THERE IS a short machine code program which will be of interest to any ZX-81 owner who has fitted the *Sinclair Projects* hi-res board to a computer. With the *Sinclair Projects* board the hi-res has to be switched-on manually with a switch fitted to the case but that can now be switched on automatically with these extra two lines of program:

Put a REM at line 1 just five characters long, e.g., 1 REM AAAAA. Then:

```
Poke 16514,62    lda,N
Poke 16515,30    N—Normal display
Poke 16516,237
Poke 16517,71    ldi,a
Poke 16518,201   ret
```

The number poked into address 16515 determines whether hi-res is switched on or off. Use Poke 16515,30 for a normal display and poke 16515,62 for hi-res display.

To call the short machine code program put this extra line in your main program: 2 RAND USR 16514.

The *Sinclair Projects* hi-res board is wired into the ZX-81 at address 15872.

R F Daw, Kingswinford, West Midlands.

Design guide

IN MY article Design Guide in *Sinclair Projects* October/November 1983, several slight errors have crept in as follows: page 37, line 7540 "SP" should be "sp"; line 7620 has "P" p) should be missing; line 7630 "isp" should be "2*isp"; line 7720 "go" should be "io".

M Farnsworth, Bournemouth.

SHOPPING LIST

Project buyers' guide

HERE IS a list of suppliers for difficult-to-obtain items which have been used in projects.

PCB mounting 3.5mm. jack sockets as used in the Central Heating Controller project.
MS Components Ltd

Edge connectors 23-way for ZX-81 and 28-way for Spectrum.
Innovonics

Extender cards for fitting to rear of edge connector to allow stacking add-ons.
23-way for ZX-81 ZXTONGUE
28-way for Spectrum SPECTONGUE
Innovonics

AY-3-8910 Sound Chip
Cricklewood Electronics
Watford Electronics

Weather Station anemometer
Ribbon cable
DIL headers
Innovonics

Replacement tape heads
Maplin Electronic Supplies Ltd

MS Components Ltd, Zephyr House, Waring Street, West Norwood, London SE27. Tel: 01-670 4466.

Ambit International, 200 North Service Road, Brentwood, Essex. Tel: 0277-230909.

Watford Electronics, 33-34 Cardiff Road, Watford, Herts. Tel: 0923-40588.

Innovonics, 147 Upland Road, East Dulwich, London SE22.

Cricklewood Electronics Ltd, 40 Cricklewood Broadway, London NW2 3ET. Tel: 01-452 0161.

Shudehill Supply Co Ltd, [redacted] Shudehill, Manchester M4 4AW.

Reading listings made much easier

READER'S TIP

RAYMOND ROSE of Devizes, Wiltshire sends a good idea which could help when reading listings printed on the Sinclair printer. It needs only an unused cassette case and a piece of twin card or paper 105 × 68mm. cut as shown in the diagram.

Rose suggests printing on the card *Sinclair Projects ZX-printout line reader* but we are more modest and will leave the wording to readers.

Attach the card to the inside of the cassette cover lid. Break off the two protruding lugs from the base of the tray. With the case opened, lid to the right, thread the printout, face-up and from below, through the hinge aperture. Close the lid and the printout will be trapped sufficiently to enable it to be drawn through, one line at a time, without slipping. The reader can be hand-held or attached to the monitor with a sticky pad.

For this simple idea, Rose will receive our special fee of £10. Other readers who have similar good ideas should sent them to *Sinclair Projects*, 196-200 Balls Pond Road, Islington, London N1 4AQ.

Cut

Cut

Playing Boole boosts circuit understanding

In the second of his series of articles to help explain the theory behind many of our projects, Joe Pritchard considers Boolean Algebra, more logic functions and the Schmitt trigger circuit.

IN THE LAST issue I looked briefly at logic gates and logic integrated circuit families. This time I consider Boolean Algebra, some more logic functions and the Schmitt trigger circuit.

Like the rest of electronics, which utilises many mathematical equations to describe the behaviour of circuits, digital electronics has at its disposal a means of describing how logic circuits work. This branch of mathematics, devoted to digital logic, is called Boolean Algebra, after its creator, George Boole. It was formulated long before the advent of computers and so Boole had no idea to what use his creation would be put.

There are variables as in normal algebra. They are called logical variables and may assume one of two values — 0 or 1. That should not surprise us as we are familiar with the two-state system from the previous article, which explained some of the basics of digital logic. 0 is said to be the complement of 1 and vice-versa. We can write this using symbols as;

$$\overline{0}=1 \qquad \overline{1}=0$$

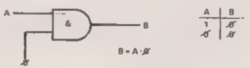
We can also say that;

If $A=1$
$\overline{A}=0$

Here we have used a simple logical variable and have assigned a value to it. It should be noted that the variable $\overline{A}$ is pronounced as "A bar" or as "Not A". The act of complementing a logical variable can be seen by inspection of the foregoing to be the same as inverting the value of the variable, using the logical invert function we met last time, i.e., a variable with the value of 1 becomes 0 and a variable with value 0 becomes 1.

Before proceeding further, let us say for what we use Boolean Algebra. It is to help us analyse what kind of function an already existing logic circuit performs and to help decide whether such a circuit can be simplified. We also use it to design logic circuits, given that we know what kind of logic function we wish the circuit to perform on its input signals.

Figure 1.



In Boolean Algebra, a series of general rules are called Boolean Postulates. It is around those rules that the algebra revolves and they are shown in table one. In the table, the "+" symbol indicates the inclusive OR function and the "." symbol indicates the AND function.

If we replace one of the constants with a variable, which can be either 0 or 1, we can draw the conclusions in table two.

They are called Boolean Theorems. If we consider table one further we can say that the following is true — remember that $\overline{0}$ is 1:

$$\left.\begin{array}{l} X\cdot Y = Y\cdot X \\ X+Y = Y+X \end{array}\right\} \text{Commutative Property}$$

Those algebraic properties, as they are known, are the same as those which exist in normal arithmetic. A second is called the Associative Property and the following equations show that. The final property to be considered is called the Distributive

Figure 2.

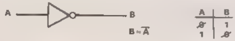
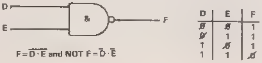


Figure 3.

Figure 4.



$F=\overline{D\cdot E}$ and NOT $F=\overline{D}\cdot\overline{E}$

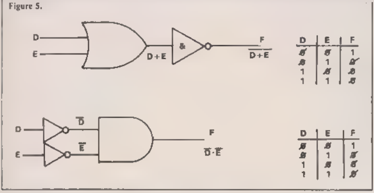


Figure 5.

Property. It is the same as the conventional arithmetic process of factoring a bracketed expression.

Associative property

$A+(B+C)=(A+B)+C$

$A\cdot(B\cdot C)=(A\cdot B)\cdot C$

Distributive property

$A\cdot(B+C)=(A\cdot B)+(A\cdot C)$

It is possible to demonstrate the properties and theorems by using simple logic circuits. For example, figure one shows how we might demonstrate the theorem $A\cdot 0=0$:

Similarly, figure two shows the circuit for demonstrating the complementing of a variable.

Examining the idea of complementing a variable twice, we find that if we perform the function on a variable twice, the variable is left with the same value with which it started. Thus we can state a new theorem;

$\overline{\overline{A}}=A$

We could prove that by building the circuit shown in figure three. So far, the properties and theorems at which we have looked have followed from the rules set out in table one. We will now view some of the more powerful rules of Boolean Algebra, which are slightly more difficult to follow. For our purposes, which are to put them to work in logic design, suffice to say that they work and that they can be shown to work by the use of truth tables and logic circuits.

The first of the theorems we shall consider are called DeMorgans Theorems. There are two of them and they are commonly called DeMorgans NAND theorem and DeMorgans NOR theorem. They are written algebraically as:

First Theorem — NOR

1. $\overline{D+E}=\overline{D}\cdot\overline{E}$

Second Theorem — NAND

2. $\overline{D\cdot E}=\overline{D}+\overline{E}$

The bar, when over both variables in an expression, as shown, means that you first perform the AND or OR operation and then take the complement of the result of that operation. That is the type of situation which exists in relation to the NAND function we encountered last time — figure four.

Note that DeMorgans Theorems link the concepts of OR and AND mathematically, thus indicating the possibility of creating the logical OR function from the logical AND function and vice versa. That confirms what we found by experiment in part one of the series, when we did that using logic gates. Now let us examine these two theorems in more detail using logic gates and truth tables.

Considering $\overline{D+E}=\overline{D}\cdot\overline{E}$, if that is true we should be able to construct two circuits to perform the logic operations in the foregoing expression on the variables D and E. One circuit will perform the function $\overline{D+E}$ on the variables and the other will perform the function $\overline{D}\cdot\overline{E}$. If the theorem is correct, the truth tables for both circuits will be the same.

Remembering part one, we can try the circuits shown in figure two. Breadboard them if you wish, using a 7432 for the OR gate, a 7404 for the inverters and a 7408 for the AND gate. Details of how to monitor the outputs from the circuits can be found in part one.

Using similar techniques, it is possible to repeat the experiment for De-

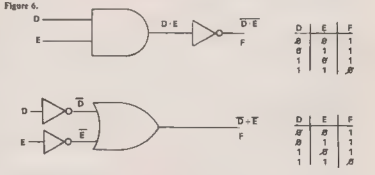


Figure 6.

Table 1.

a $\overline{1}=0$	e $\overline{0}=1$
b $0\cdot 0=0$	f $0+0=0$
c $1\cdot 0=0$	g $0+1=1$
d $1\cdot 1=1$	h $1+1=1$

Table 2.

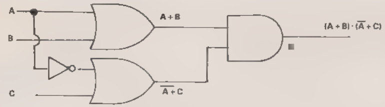
$A\cdot 0=0$	$A+0=A$
$A\cdot 1=A$	$A+1=1$
$A\cdot A=A$	$A+A=A$

A	B	OUT
0	0	1
0	1	0
1	0	0
1	1	1

Label out to be C

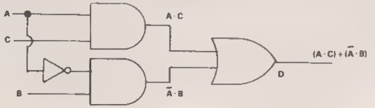
DIGITAL ELECTRONICS

Figure 7.



A	B	C	■
0	0	0	0
0	0	1	0
0	1	■	1
0	1	1	1
1	0	■	0
1	0	1	1
1	1	0	0
1	1	1	1

Figure 8.



A	■	C	D
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Morgans second theorem, using the circuits shown in figure six.

In both examples, note the identical behaviour of the two circuits in each case.

The final theorem we need to examine before we can study how we can use Boolean Algebra practically is called the Law of Consencus. It is another of the theorems best proved by using logic circuits and truth tables. The Law of Consencus states:

$(A+B)\cdot(\overline{A}+C)=A\cdot C+\overline{A}\cdot B$

From that equation, we can develop two circuits in the same way. The circuits are shown in figures seven and eight. Note again the fact that the truth tables are the same.

One concept which has become apparent from the examinations of theorems is that we can have different logic circuits to perform the same logic function. That was demonstrated in a practical fashion in part one. There we constructed and AND function from an inverter function and a NAND function. Analysing this using Boolean Algebra:

$\overline{A\cdot B}=C$

$\overline{\overline{A\cdot B}}=A\cdot B$

$\therefore\ A\cdot B=C$

Thus we get the function AND. After a little practice, it becomes possible to build circuits to perform a given logic function using only a logic equation and Boolean Algebra. We shall now examine a simple example of how it is done.

The process of synthesising a logic circuit from scratch goes via a verbal or written description of what we want the device to do, to a truth table which results from this description, then to a logic equation and, finally, to the circuit.

If we want to design a logic function to give a high output when both of its two inputs are the same, it is easy to see that the truth table for this function is that in table three.

The next step is to list the circumstances which give a high output in the form of Boolean equations. When both A and B are equal to 0 we get a high output and when both A and B are equal to 1 we get a high output. So how can we write those facts in the form of equations? First, we treat any input with a value of 0 as a NOT-ed term and any input with a value of 1 as a non-NOT-ed term in the equation. So we can write:

$\overline{A}\cdot\overline{B}=C$

$A\cdot B=C$

Either one of those conditions arising at the input will give rise to a high state at the output. Thus we can relate those two equations by use of the OR expression:

$(\overline{A}\cdot\overline{B})+(A\cdot B)=C$

We then have a logic equation which describes the function we wish

Table 4.

$D=\overline{A}$

$F=\overline{C}$

$E=D\cdot B$

$G=E+F$

$\therefore G=(\overline{A}\cdot B)+\overline{C}$

A	B	C	G
0	0	0	1
0	0	1	■
0	1	0	1
0	1	1	1
1	0	0	■
1	0	1	1
1	1	0	0
1	1	1	1

Table 5.

TTL	7413	DUAL	4 INPUT NAND
	7414	HEX	INVERTER
	74132	QUAD	2 INPUT NAND
CMOS	4093	QUAD	2 INPUT NAND
	40106	HEX	SCHMITT TRIGGER

Table 6.

	CMOS	TTL
V·	4.3	2
V·	0.7	0.6

Figure 9.

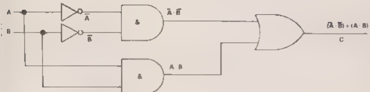


Figure 10.

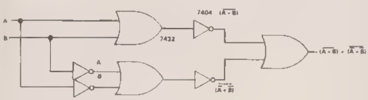


Figure 11.

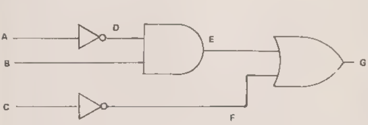


Figure 12.

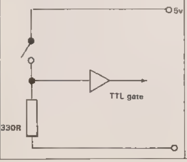
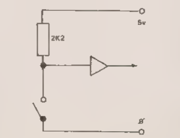


Figure 13.



to implement. We can see if the equation is correct by substituting into the expression in the way normal algebraic expressions are checked.

Now, to the circuit. If we look at the tables of postulates we can convert this equation into a logic diagram. Figure nine shows that and it also shows how the values of the variables A and B are modified on their passage through the circuit:

Practically the circuit can be built using two gates from a 7404 chip, two from a 7408 chip and one gate from a 7432 chip. We have already seen that there are usually more methods of synthesising a logic function than usually are apparent. Let us see if we can find another say of performing this logic function but with the use of fewer logic devices. We will be using some of the more complex Boolena theorems:

$$C = (\overline{A} \cdot \overline{B}) + (A \cdot B)$$

Using DeMorgans First Theorem;

$$(\overline{A} \cdot \overline{B}) = (\overline{A + B})$$

$$C = (\overline{A + B}) + (A \cdot B)$$

Using DeMorgans Second Theorem;

$$(\overline{A \cdot B}) = (\overline{A + B})$$

and noting the fact that what we do to one side of an expression we must do to the other side to maintain equality, and remembering figure three, we will NOT both sides of the expression.

$$(A \cdot B) = (\overline{\overline{A} + \overline{B}})$$

$$\therefore\ C = (\overline{A + B} + (\overline{\overline{A} + \overline{B}})$$

The significance of the bar over both of the functions in the brackets is that we carry-out the function in the brackets on the variables first and then we invert the result. The resulting circuit for this expression is in figure 10.

We are now using only OR gates and inverters and that reduces the chip count needed to build the circuit. The chip count is the number of integrated circuit packages required to build the circuit. In figure 10 we have a chip count of two rather than the three of figure nine. That reduction in chip count obviously is useful where we require a circuit small in size.

The circuit could have been implemented using an exclusive OR chip and an inverter but if the OR circuits were already on the circuit board

DIGITAL ELECTRONICS

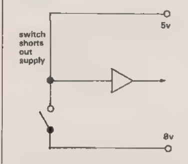


Figure 14.

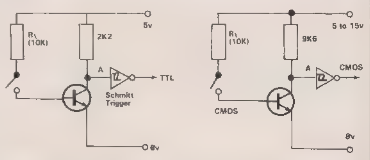


Figure 15.

being made it would make sense to use them.

It is hoped that this example has shown how we can use Boolean Algebra to design a logic circuit and how we can use it to simplify already existing ones in some cases.

The simplification procedure leads us to the second use of Boolean Algebra, to analyse circuits to simplify them or to find what logic function they perform on their inputs. That is effectively the reverse of the procedure we have just examined. Let us analyse the circuit of figure 11.

The first task is to label the inputs and outputs of each gate in the logic network. We then evaluate the logic functions for each point on the logic circuit diagram we have labelled.

Thus we have now worked out a logic equation for the circuit, describing the output G in terms of combination of the inputs A,B,C. The final stage of the analysis is to draw-up ■ truth table. We do so by substituting the values 1 and 0 into the equation

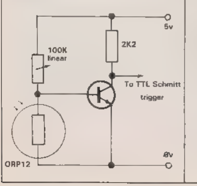


Figure 17.

instead of A, ■ and C. Take for example, the situation when all three input variables are equal to 0:

$G = (\overline{0}\cdot 0) + \overline{0}$

$\therefore G(1\cdot 0) + 1$

$\therefore G = 0 + 1$

$\therefore G = 1$

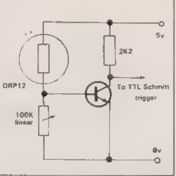


Figure 16.

The truth table we obtain for the circuit is shown in table four. With logic systems with many inputs, there are obviously different ways in which the truth table can be drawn, depending on how we label the inputs.

We will now study some techniques a little more closely regarding the practicalities of digital electronics — the realities of connecting the TTL and CMOS gates to the outside world.

Last time we looked briefly at how to input signals to a TTK gate. Now we look at the problem in greater detail. If we take a typical TTL gate we find that the input voltage level for a low input to be recorded, V_L, must be less than about 0.8 volts. The voltage needed for a high input to be recorded is called V_H and is any voltage between about two and five volts.

Figure 12 shows a method of interfacing a switch to a TTL input in such a way that when the switch is closed, the TTL gate sees a high input. The 330R resistor is to ensure a low input on the gate when the switch is open. It is called a PULL DOWN resistor.

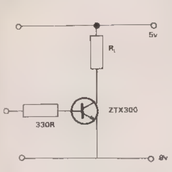
The circuit in figure 13 gives a low input when the switch is closed. The resistor supplies a high signal to the gate until the switch is closed. If you refer to part one you may remember that an unconnected TTL gate will assume the high value naturally. It is good design practice, however, to tie all loose ends when working in a field so intricate as digital electronics.

The resistor in figure 13 is to pull the TTL input up to a high level and hence is called a PULL UP resistor. If we are utilising CMOS devices instead of TTL, we can use the same resistor values in the circuits. As a rule of thumb, pull up resistors should be between 1K and 70K for TTL inputs and between 2K2 and 1M5 for CMOS inputs.

Pull down resistors should be between 100 and 500 ohms for TTL inputs and between 100 ohms and 1M5 for CMOS. Note that direct connection between the gate and either ground or 5V, omitting the resistor, is not possible, because as soon as you depress the switch, the power lines are short-circuited — figure 14.

It may occur occasionally that we wish to drive the input of a TTL device from a non-TTL electronic device. Some devices have what are

Figure 18.



known as TTL level compatable inputs and outputs, which means that the device sees voltage inputs on it in the same way as TTL devices and gives outputs of the same voltage levels as the TTL 0 and 1 voltage levels.

The Z-80 processor at the heart of the ZX series computers is such a device but not everything is like that. Two typical devices we need to treat specially are transistors and CMOS devices.

I would like to examine the problems associated with transistors and in part three we will look at interfacing CMOS devices to TTL devices.

With regard to interfacing transistors to the inputs of TTL and CMOS gates, the two circuits in figure 15 can be used. The transistor should be of reasonable quality. Typical devices to use are the ZTX300 or any high-gain fairly fast NPN switching transistors.

Note that depressing the switch will cause a logic condition 0 at point A on the circuit. That will result in a logic 0 being applied to the CMOS gate and a logic 1 will be applied to the input of a TTL gate connected to the output of the Schmitt trigger. The latter device ensures a good switching signal to the TTL gate connected to its output and will be described later. The value of R_X is typically about 10K.

You might like to try connecting a sensor in place of the R_X. A light Dependent Resistor — LDR — would give the circuit shown in figure 16. One point to note about the use of transistors in that way is that the typical voltage across the collector-emitter circuit of the transistor when current is flowing through it is about 0.6 volts. That is the situation which exists when the switch is depressed and that voltage is a little close to the upper limit of the 0 condition for a TTL gate.

That is why a high gain switching transistor is needed. The potentiometer shown in figure 16 enables you to adjust the sensitivity of the circuit. As the light level changes, it alters the resistance of the LDR. As it rises, the resistance falls. Thus you can set the brightness level at which the transistor switches. To get a circuit which switches as the light level falls we use figure 17.

As well as these simple circuits

Figure 20.

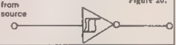
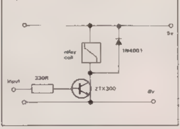


Figure 19.



there are more advanced techniques of interfacing transducers to logic circuits which use other integrated circuit packages instead of transistors and they will be looked at in a future article.

Transistors are also used to switch output devices. When used in that role they enable the switching of higher currents — figure 18.

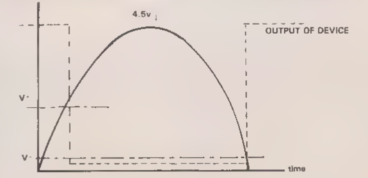
R_L is the load of the transistor circuit and should take no more than 40–60mA of current when turned on. That is still a considerable improvement on the current available when a TTL gate is sinking or sourcing current. The load could be a light bulb, a loudspeaker or a relay. The latter would enable the logic circuit to switch even higher currents and voltages. The circuit shown in figure 19 will switch a relay when a logic 1 is applied to the input. The diode is to protect the transistor from damage when the circuit switches off the relay.

One thing to note is the concept of bypassing. It is a technique in which we connect a capacitor of, say, 47μF or so across the power lines of every second chip. The capacitor should be wired close to the chip in question. Its role is to dispose of any electrical noise reaching the power lines of the circuit from the switching action of the TTL or CMOS gate.

The noise could possibly cause trouble and so these capacitors pass all the noise to the 0 volts line where it can do no harm. Devices such as relays and motors should have, as well as the diode, a larger value capacitor across them. A typical value is about 0.1 to 0.47μF.

It is useful to consider in detail a device we have already considered in

Figure 21.



passing, the Schmitt trigger, which can be described technically as a device which changes its state only whenever the voltage on its input goes beyond a certain value. That value is different for positive-going and negative-going voltages. A typical TTL Schmitt trigger is the 7414 device which has a built-in inverter function. The symbol for it is shown in figure 20.

Figure 21 shows what happens to the output when we apply a changing voltage to the input.

V^+ is the voltage at which the threshold is reached which will cause a change in the output state from 1 to 0 in this case. It is usually about two volts. V^- is the voltage at which we get a reversion from the 0 state to the 1 state. That is typically about 0.6 volts for a TTL device. Thus the 7414 gives a TTL low output signal for input voltages above two volts and TTL high output signal for input voltages below 0.6 volts. The voltage range between 0.6 and two volts is called the Hysteresis Interval.

If we examine figure 22 we see one of the main uses for Schmitt triggers in cleaning noisy signals, such as those which might result from being passed along a long run of cable in an electrically noisy environment.

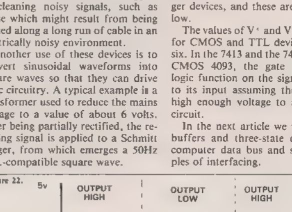
Another use of these devices is to convert sinusoidal waveforms into square waves so that they can drive logic circuitry. A typical example is a transformer used to reduce the mains voltage to a value of about 6 volts. After being partially rectified, the resulting signal is applied to a Schmitt trigger, from which emerges a 50Hz TTL-compatible square wave.

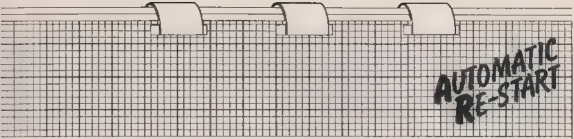
There are several TT and CMOS packages which contain Schmitt Trigger devices, and these are shown below.

The values of V^+ and V^- are shown for CMOS and TTL devices in table six. In the 7413 and the 74132 and the CMOS 4093, the gate performs a logic function on the signals applied to its input assuming they are of a high enough voltage to activate the circuit.

In the next article we will look at buffers and three-state devices, the computer data bus and some examples of interfacing.

Figure 22.





Making sure you do not lose control

In many of the serious uses of Sinclair machines it is important that programs do not stop running if there has been a power stoppage. Raymond Hopkins reports

IMPLEMENTATION of an autoboot facility on the ZX-81 is an invaluable feature normally found only on expensive, professional computer/controllers. It provides for the automatic execution of user-prescribed software on power-up. The most significant advantage of autoboot is that, where the computer is being used as a controller, it allows ordered recovery from a power failure of whatever duration.

Such an ability is vital in any serious control application, of course; without it, even a momentary hiatus in the mains supply would suffice for control to be lost, possibly with disastrous consequences. The ZX-81 implementation described does not interfere with normal operation of the computer, relies on a standard cassette player for mass storage backup, and can be housed comfortably in the computer case with no elaborate connections. Further, it can be built from standard parts for less than £5.

Once this autoboot circuit has been constructed and installed — for a preview see figure one — preparation for its use is simple. A lead with miniature jack plugs at either end is connected between the new remote socket on the ZX-81 and the remote socket on a tape recorder. The control program should have been saved in autorun mode, several times on the same side of a fairly long tape — a C90 for example. Link the computer and recorder ear sockets, depress the play switch on the tape recorder, and all is ready. Automatically, at switch-on, the ZX-81 will set the recorder player load and run the control program and switch off the cassette motor when loading is finished. In the event of a mains failure, when power is restored the same process will be repeated automatically. What could be more straightforward?

The autoboot performs its wonders through the cunning stratagem of simulating key presses on the ZX-81 keyboard. Figure 2a illustrates the normal manner of keyboard operation from a matrix of simple key switches; individual key presses are decoded by detecting which KBD — KeyBoarD — line has been connected to which address line. Figure 2b, on the other hand, demonstrates how the same end can be achieved using an inexpensive analogue gate; a 4016 or 4066 CMOS IC contains four such switches.

With its control line at logic zero, the gate appears open circuit from either direction, in which condition it has no effect on anything connected to either signal terminal. That means that such gates may be wired in parallel with the normal keyboard switches.

When the control line is set to logic one, on the other hand, a low resistance path is created between the signal terminals. The gate behaves, in effect, like a single-pole relay but is far cheaper, more compact and easier to drive. So what the bootstrap circuit does is to connect the keyboard and address lines used when manually typing in 'LOAD " " (NEWLINE)', to the signal terminals of four analogue gates — cf, figure three, IC5. Then, at the proper time, at a manageable speed and in the correct order, it 'opens' the appropriate gates and the computer behaves as if a phantom typist were at work. Referring to the right-hand side of the circuit diagram in figure three you will notice that only three address lines and two KBD lines are used. KBD 3 connected to A14 decodes 'LOAD' (in 'K' mode); KBD 0 connected to A8 & A13 decodes ' " '; KDB 0 connected to A14 decodes 'NEWLINE'.

The rest of the circuit handles the orderly operation of the analogue

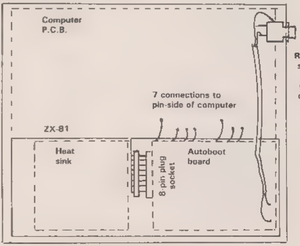


Figure 1. Installation of ZX-81 Autoboot facility

gates and switches the cassette motor at the required times. C1 and R1 reset the whole autoboot circuit at power-up, holding it in that state until the ZX-81 has had sufficient time to run through its initialisation procedure. The initialisation period is longer on a 16K machine, so in that case R1 should be a 270K resistor, while on an unexpanded machine 100K will suffice.

IC3 is a decimal counter if pulses are applied to its clock input — CLK — it will set high each of its 10 outputs in turn, repeating the cycle as more pulses are received. That input is driven by an oscillator built around IC1a and IC1b and running at about 1Hz, the frequency being determined by the values of C4 and R3. C5 is included to de-couple unwanted high-frequency oscillation when the whole circuit is de-activated. With the help of the two remaining exclusive-or gates of IC1, the decimal outputs of IC3 are used to switch the analogue gates of IC5 in the correct sequence.

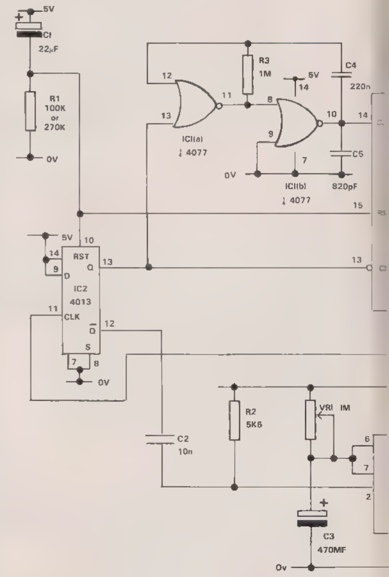
The highest and last output, Q10, clocks D-type flipflop IC2 with several results. In the first place the oscillator is disabled by the Q output of IC2 and, second, just to make sure that everything grinds to a halt, the same signal disables IC3. Finally, flipflop output takes pin 2 of IC4, a 555 timer, to zero volts, initiating a single shot — one pulse.

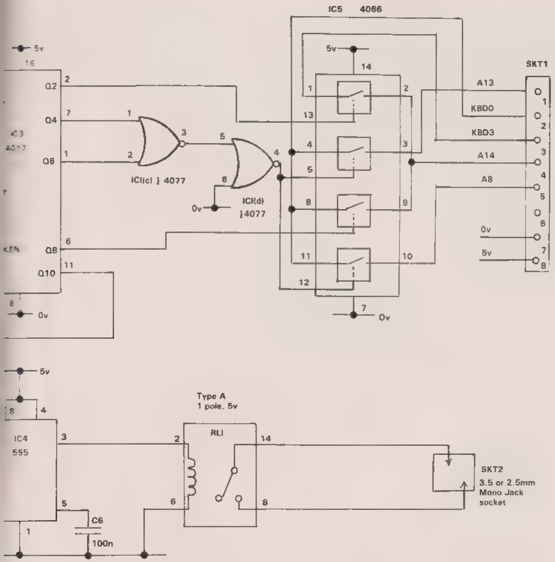
At this point, only that part of the circuit involving IC4 and RL1 is active; the rest is in a stable and quiescent state. Having performed their role, all analogue gates are open-circuit. The single-shot pulse appearing at IC4 pin 3 energises the coil of RL1 and so completes the motor circuit in the tape recorder via its remote socket.

How long the motor remains on is determined by the value of C3 and the setting of VR1. With the latter at maximum resistance the single shot will last about eight minutes.

Since the rest of the circuit remains inactive; meanwhile, IC2 Q output will still be at logic zero, the condition which triggered the single shot. You may well wonder why further pulses are not generated. Two components

Figure 3: Circuit diagram of ZX-Autoboot board




IC5 4066
5v
14
SKT1
Q2
Q4
Q6
Q8
Q10
ICl(c) ¼ 4077
ICl(d) ¼4077
0v
A13
KBD0
KBD3
A14
A8
0v
5v
7
0v
IC4
555
C6
100n
Type A
1 pole. 5v
RLI
SKT2
3.5 or 2.5mm
Mono Jack
socket

AUTOMATIC RE-START

Figure 2a: Normal keyboard operation

have been included in the line-driving IC4 pin 2 to avoid that state of affairs. C2 and R2 consitute a differentiating network which transmits the changing signal as IC2 pin 12 drives low, thus allowing the triggering of the desired pulse. C2 very quickly charges through R2, however, and prevents further activity.

Before leaping into the fray with your soldering iron, spend a little time studying all the illustrations to obtain a feel for the general assembly of the autoboot facility. It will be noted that four CMOS ICs are in the circuit design. Mere mention of CMOS brings flutters to many but there is no cause to be nervous at all. Modern CMOS is well-protected against reasonable levels of static, so the simplest of precautions should allow you to work confidently with it.

Check that your soldering iron is earthed — most of them are — and discharge any static you may be carrying through some large metal object, or a copper water pipe, before starting work. The prototype circuit was built without IC sockets. Taking the precautions outlined, the reader is unlikely to encounter difficulties in following the same course. If sockets are used, however, low profile ones would be advisable.

Construction might follow this procedure. First, cut a piece of 0.1in. pitch Veroboard to size, i.e. 20 copper strips by 27 holes long. Following figure four cut the tracks where indicated, using a spot cutter or a sharp scalpel blade. In either case, centre the cuts on the holes indicated, removing a minimum of copper to break continuity. Brush away copper swarf and check all cuts carefully before continuing.

Then improvise SKT1 by cutting in half a 16-pin, turned-pin DIL socket.

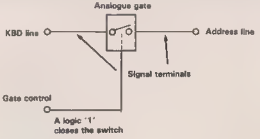
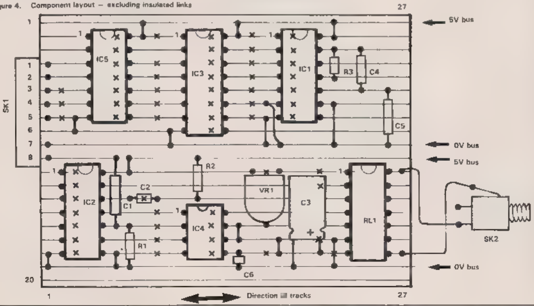


Figure 2b. Emulation of keyboard switches using analogue gates

Figure 4. Component layout — excluding insulated links



Remove pin 6 and blank-off its position to serve as a locating guide. Referring to figures four and five, solder SKT1 to the track side of the Veroboard in a horizontal position. There, as elsewhere, confine the flow of solder to the area immediately surrounding the hole being used; otherwise it could be a messy job fitting later components into the fairly compact layout.

Next solder in the IC sockets — or the ICs themselves — along with RL1, precisely according to the layout orientation in figure four followed by power links from the ICs to the 5V and 0V buses. Remember to connect each bus pair together.

Then the discrete resistors and capacitors can be added and, finally, the connections shown schematically on the circuit diagram — figure three — should be made with single-strand, insulated wire. Note that C3 has been specified as having a working voltage of 6.3V, with good reason — any higher operating voltage would involve too bulky a component.

When tackling the wiring, incidentally, it would be a good idea ■ tick each link on the circuit diagram as it is soldered to avoid confusion. To complete the board, two flexible leads should be taken from pins 14 and 8 or RL1 to a panel-mounting jack-socket — 3½ or 2½mm.

Refer then to figure five. If you have a finetipped soldering iron,the second half of the turned-pin DIL socket used to construct SKT1 might serve for PL1, wires being soldered into the wells of the socket. That is a fairly delicate operation, so the less experienced will possibly prefer to use half of a 16-pin DIL header plug. In either case, remove pin 6. That done, take off the bottom half of the ZX-81 casing. Leaving the computer PCB in the upper half of the case with keyboard attached, wire PL1 in accordance with figure five, using flexible, stranded leads each about 4in. long.

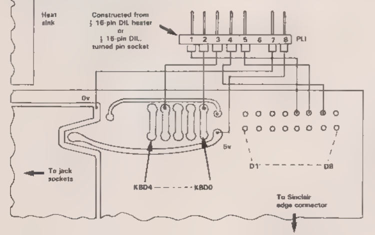
The final step is to drill a hole to accept the autoboot jack socket in the upper casing of the ZX-81, on the right-hand side and near the back of the computer. It should be unnecessary to remove the computer PCB from the upper moulding. A piece of light, non-conductive foam, spot-tacked into the front right of the lower moulding, would provide a secure seating for the autoboot board in its final position.

Before installing the circuit and connecting it, if you have not already done so, check your work carefully. If you have a multimeter, set it to a low ohms range and ensure that no short exists across the power rails of the board 5V to 0V. If all is well, fix the jack socket into its mounting hole, connect PL1 to SKT1, and seating the autoboot card comfortably on the foam as you lower the upper half of the computer case on to the lower, ensure that no part of the circuit can short to the heatsink as you do so.

Switch on and after a short delay, 'LOAD "" ' should appear on the screen, disappearing quickly as 'NEWLINE' is pressed by the bootstrap circuit. If nothing happens, switch off and double-check everything carefully. Should an incorrect command appear on the screen, verify the wiring of IC3,IC5,PL1 and SKT1. If everything has performed as planned, connect a lead with appropriate plugs between the new remote socket on the computer and the recorder remote socket.

Power-down the computer, engage the 'PLAY' switch on the recorder, switch on the ZX-81, and verify that the recorder starts playing at the end of the LOAD command sequence. All that remains is to adjust VR1 so that the tape recorder is switched on for ■ period appropriate to the size of your bootstrap program. There is no reason, in principle, why the autoboot circuit should not work with a Spectrum, though I must confess that I have not tried it. The wiring from PL1 would, of course, need to be modified to suit the layout of the Spectrum PCB.

Figure 5: View of pin side of ZX-81 PCB (X2) showing points connecting ■ PLI



PARTS LIST

IC1	4077
IC2	4013
IC3	4017
IC4	NE555
IC5	4066
RL1	Form A,1 pole, 5V e.g., Maplin FX88V RS 349-383, Ambit 46-61500
VR1	1M lin, submin., e.g., Ambit 48-10501
C1	22uF 16VW or less
C2	10nf
C3	470uF, 6.3VW e.g., Ambit 05-47705
C4	220nF
C5	820pF
C6	100nF
R1	100K 1K ZX-81 — or 270K — 16K ZX-81
R2	5K6
R3	1M
	All resistors ¼ watt or less.
PL1	16-pin, turned-pin, DIL socket
SKT2	3.5mm or 2.5mm, panel mounting, mono jack socket.
Veroboard	0.1in. pitch, 20 tracks by 27 holes.

WEATHER STATION 2

A vane attempt to try and catch the wind

In the second of his series of articles on how to build a simple weather station controlled by the Spectrum, Graham Bradley shows how to build a device to measure wind direction

WE CONTINUE the series on the ZX weather station with details of the construction of a wind direction indicator. A primitive form of the device is to be found mounted on the top of many church spires. When a flat vane is pivoted off-centre it will turn so that the biggest area of the vane is downwind. The vane need not be rectangular; some possible shapes are shown in figure one. The vane should be counter-balanced to reduce turning moments about point X.

The remote measurement of wind direction involves the use of a coded disc. If a transparent disc is divided into segments, each segment can have areas blocked-out so that each position of the disc has a special code associated with it. That is the principle behind many rotary or shaft-encoding devices. The pattern in each segment is detected using photo-transistors, photo diodes or photo-resistors. The most obvious code to use would be the binary code and in this situation it will work satisfactorily.

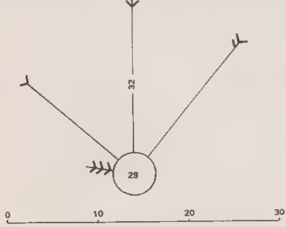
An alternative code which is more commonly-used in practice is the Gray code, named after the man who developed it to overcome the problems of 'race' which can occur in more than two bits of a counter or logic circuit change at once. There is a possibility that one bit will change slightly before the other because of slight misalignment of the code detectors or because of propagation delays through the logic gates.

In the transition from binary eleven to binary twelve it is possible that bit 2 may change before bit 1 so that an illegal pattern will exist momentarily — see table two.

In going from one number in the Gray code sequence to the next number, only one bit position changes value, so eliminating the problem. There are other patterns which will perform the same function; you may like to try and discover them.

The four-bit Gray code arranged to form the pattern on a shaft encoder is shown in figure three. With the four-bit code a resolution of 22.5 degrees is possible, which is adequate for our purposes. A five-bit Gray code will give 11.25 degrees resolution if you require it. The sequence for a five-bit code is to be found in many textbooks

Figure 4. Wind rose.



This wind rose should be read:

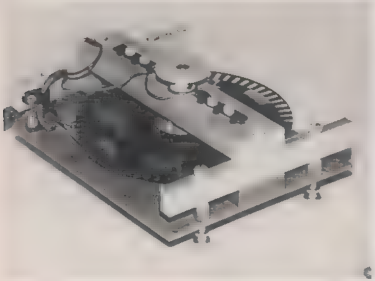
From N: 32 percent, force 4 — From W: 1 percent, force 6

From NE: 20 percent, force 3 — From NW: 18 percent, force 2

Calms, light airs, and variables, 29 percent.

Table 1. Beaufort Scale with corresponding wind values.

Beaufort Number	Knots	Miles per hour	Speed Symbol
0	0	0	
1	1-3	1-3	
2	4-6	4-7	
3	7-10	8-12	
4	11-16	13-18	
5	17-21	19-24	
6	22-27	25-31	
7	28-33	32-38	
8	34-40	39-46	
9	41-47	47-54	
10	48-55	55-63	
11	56-63	64-73	
12	64-71	74-83	



Commercially-available 5-bit encoded disc, suitable for connection to weather vane.

General view of a simple weather vane.

or you can work it out by observing the pattern of ones and zeros in the four-bit code and extending it.

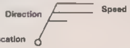
The first column starts with a zero and then alternating groups of two ones and two zeros occur as you go down the columns. The second least-significant-digit column starts with two zeros and then has alternating groups of four ones and four zeros. The third column starts with four zeros and has alternating groups of eight ones and eight zeros. An eight-bit code produced in this way will have a resolution of one part in 2^8 or 1.4 degrees.

Photo-transistors and diodes from

Table 2.

	Binary	Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
[illegible]	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Table 3. Wind direction and speed symbols.



Wind Direction:

Wind from N	Wind from S
Wind from NNE	Wind from SSW
Wind from NE	Wind from SW
Wind from ENE	Wind from WSW
Wind from E	Wind from W
Wind from ESE	Wind from WNW
Wind from SE	Wind from NW
Wind from SSE	Wind from NNW

Wind Speed:

Knots	mph	Knots	mph
0	[illegible]	33-37	38-43
1-2	1-2	38-42	44-48
3-7	3-8	43-47	49-54
8-12	9-14	48-52	55-60
13-17	15-20	53-57	61-66
18-22	21-25	58-62	67-71
23-27	26-31	63-67	72-77
28-32	32-37	68-72	78-83

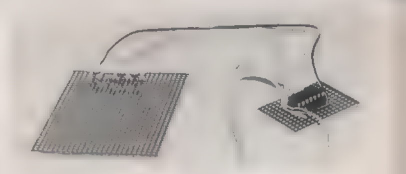
WEATHER STATION 2

various sources were used in the prototype. The only change required is that the load resistor will have a different value, depending on the light current of the device used. With the light source used the collector voltage of the photo-transistor should fall below 2V when illuminated, so that it effectively operates the Schmitt buffer. The outputs of the Schmitt buffer can be fed to an input port on the computer.

For the prototype the four-code outputs and the pulse output from the anemometer were connected through a single buffer IC to five bits of an input port. If the port is bi-directional, one bit of it can be used to drive the watchdog circuit described previously, so that the weather station can be disconnected from the TV and left running, or the TV switched to another channel.

Wind direction is always reported as from a direction — a north-east wind is one blowing from the north-east. The wind arrows on weather maps use the shaft to indicate direction and the feathers to show wind speed. Wind velocities are sometimes reported in terms of the Beaufort Scale numbers as shown in table one.

Accumulated wind data for all the oceans is available in a series of Pilot Charts. There is a chart for every month of the year. For each five-degree square on the chart there is a wind rose which provides graphically a detailed history of wind directions and speeds for that specific area and month.

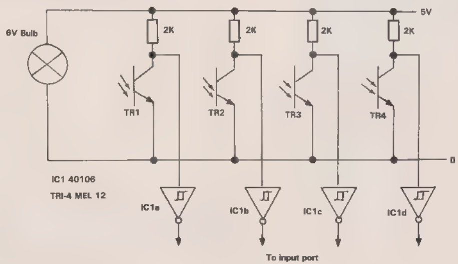


Electronics required to produce a wind direction indicator.

To produce a wind rose for your area, collect data for one month. A wind rose has the wind percentage concentrated on to eight points of the compass, so angle the base of the wind direction indicator so that each point straddles two of the 16 directions as shown in figure three. The length of the arrow for each direction will show the percentage of time for which the wind has been blowing from that direction during the month. The feathers on the end of the arrow show the average force of the wind on the Beaufort Scale.

The figure in the centre of the circle gives the percentage of calms, light airs and variable winds. When the arrow is too long to be shown conveniently, the shaft is broken and the percentage is indicated by numerals. When the arrow is too short to fit all the feathers, the feathers are shown beyond its end — see figure four. Comparisons of the wind roses for each month will show variations.

Figure 2. Wind direction circuit diagram.



WEATHER STATION 2

5-bit Grey coded disc.

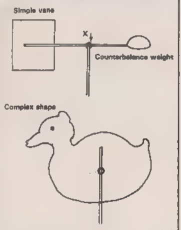
Figure 3.

Alignment of wind direction indicator when only eight points are to be read:

e.g. N – 1000 and/or 0000

e.g. NE – 0001 and/or 0011

e.g. E – 0010 and/or 0110



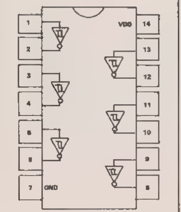
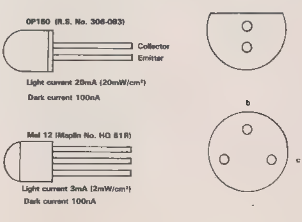


Figure 5. Pin out of 40106



TAPE HEAD RENEWAL

Heading away from the usual worn-out problems

There are many reasons for poor saving and loading of cassette tapes. One of them is the condition of the tape heads. Charles Barnatt shows, simply, how to fit new heads to your recorder.

ONE OF THE major problems experienced by owners of Sinclair and other computers is the successful recording and retrieving of programs and data on cassette tape. As computers offer no control to aid in loading or saving, all adjustments for reliable cassette operation must lie with the tape recorder.

Although the settings of the volume and tone controls are critical, the condition of the tape heads on the recorder is equally important. The volume and tone controls may determine whether the computer will load the program from the tape but it is the tape head which make the cassette output available in the first place.

Tape recorders have one or two heads to record, play back and erase information — sounds — on cassette. Sound is stored by passing varying currents through an electromagnet in contact with a magnetic tape. Those currents produce varying magnetic fields in the electromagnet — or cassette head — which, in turn, cause iron particles on the plastic tape to become magnetised to varying degrees.

To read back those signals a small, steady current is passed through the head. The magnetised particles on the tape cause fluctuations of the current, which are sensed and amplified. The erase head, which is also the record and playback head on cheaper recorders, is another small electromagnet used to set the level of magnetism of the iron particles on the tape to zero.

The tape heads must be in good condition if they are to work properly, especially for computer use when the accuracy of recording and reproduction is so critical. A head in good condition has a clean, unworn surface. Worn or dirty heads produce distorted and/or faint signals, which cause programs to load with data errors or not at all. If the heads are dirty, the correct use of one of the many available cassette head cleaner kits will solve the problem. If the heads are worn, they will need replacing.

Worn heads will have lost the sharp edges round their centre portions — figure one — and will have a dull brown and tarnished surface. As computers use higher frequencies than are used normally in audio reproduction, and because the higher frequencies are first to be affected when heads wear, heads may be worn even if little difference in audio playback has been noticed.

So if you have a recorder which has stopped working properly with your computer and head cleaning does not solve the difficulty, you can save the cost of a new recorder for the price of a new set of tape heads.

The replacement of cassette recorder heads can be summarised in the following steps:

Buy the correct head(s) for your recorder; dismantle and clean the recorder; replace the heads; align the new heads for optimum signal response; re-assemble the recorder.

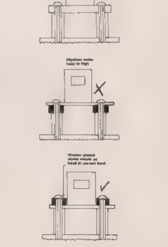
To replace the heads ■ small-tip soldering iron, screw-drivers and pliers will be needed.

To buy the correct replacement heads, first you should check whether your recorder has one or two. If it has one, a universal record/playback/erase head should be bought. If it has two, a record/playback head and a separate erase head will be needed. In the latter case be careful to buy a mono record/playback head — virtually any erase head will do.

If you can see any reference numbers on the present heads you can check easily if you are buying the correct replacements. If the numbers on the replacement do not match, do not worry; if the sizes are the same you should have no difficulties. You are most likely to obtain heads from mail order companies.

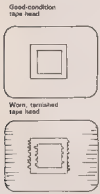
To dismantle and clean, work on a flat surface such as a table or tray and remove all flexes, cassettes and batteries from the recorder. Locate and remove the screws holding together the recorder case and part the two

Figure 2.



TAPE HEAD RENEWAL

Figure 1.



halves. If they do not part easily, check for any screws which have not been removed — such as in the base of the battery compartment — and, if that fails, try moving the case in a different way, e.g., sliding instead of pulling.

As the recorder is taken apart, be careful not to damage wires connecting components to the case — e.g., jack connectors, condenser microphone, speaker. The wires connecting them are usually short and only just allow the recorder to come apart. If the wires are very short you may have to de-solder them at one end to allow the recorder to come apart.

Do that carefully, place the soldering iron tip on the connection, pull away the connecting wire, and remove the iron tip as soon as the wire comes away. Make notes where all disconnected wires came from. Taping labels to the wires can be helpful.

Once the case is open and in two pieces it is time to locate the tape heads. Most manufacturers assemble their recorders with the controls/cassette opening, and thus the heads, on the opposite side of the chassis to that exposed when the recorder comes apart. That means you have more screws to undo, so that the whole cassette mechanism can be removed.

Usually external controls such as volume knobs have to be removed before the works come out. Those are usually push-fit and will just lever off with a srewdriver, or have a small screw in their base which needs to be released. Once they are removed and all screws undone, the recorder chassis should come out. Take the same precautions as before and be even more careful to watch for connected wires.

When all that is done you should have easy access to the heads; putting the recorder into the play position will make them even more accessible. You will almost certainly find dust and dirt, which should be removed with a soft cloth or fine brush.

The moment has then arrived to replace the heads. If the recorder has a separate erase head replace it first, as it is easier to fit than the other one. Remove the screws holding the heads, together with any washers, taking careful note of where they came from. Then fix the new head in position just like the old one.

I found that two new washers were necessary to hold it solid, as the new head had a thin metal mount and the old one had a thicker plastic one. For placement of the washers see figure two.

Then de-solder one wire from the old head and re-solder it on to the corresponding terminal on the new head. Be careful not to keep the iron tip on the terminal for more than about five seconds. Next transfer the other wire from head to head. Using that method there is no possibility of incorrect connections, because not more than one wire is disconnected at the same time. Never touch the surface of the head with a metal object as it can damage it.

The main head will be spring-mounted on one side so that its angle to the tape can be adjusted. Therefore when removing the head, be careful that the spring assembly does not fly into the air. I suggest removing the spring assembly first, taking care to note the order of any washers and the spring.

Next take off the other screw, remove the head, and replace it with the new one. First fit the solid side and then the spring-loaded one, tightening down the spring until the head looks horizontal. As before, transfer the wires from head to head, one at a time.

Cassette heads must be aligned so that they are parallel to the tape. To align the heads one side of the mouting is fitted above an adjustable spring mechanism. By tightening or loosening the screw above the spring, it is possible to adjust the angle of the tape head to the tape. The easiest way to do it is to listen to a tape. While the recorder is still apart, connect power to it, preferably using batteries. Put in a good-quality commerical music cassette or program cassette and listen to the sound while adjusting the screw above the spring. Adjust it until you get the sharpest sound. As you adjust the screw either way from that position, the sound will go dull and lose sparkle.

Then check that you can load programs into the computer. Place a commerical tape which you know loads properly into the recorder. Connect the computer to the recorder via the ear socket — if you have de-soldered, tape the wires from the socket to the computer lead. Place the recorder in play mode and set the volume/tone controls for normal computer use. Set the computer to load the program. If the program will not load, turn the adjusting screw slowly either way until the best loading pattern is obtained. Then re-wind the tape and check that the computer will load the program. If not, repeat the adjusting procedure until satisfactory loading is obtained. Your recorder is then working properly.

All that is then required is to re-assemble the recorder. Do that carefully, taking care not to knock the heads. Re-connect any wires which you have had to de-solder. It is advisable to check the recorder again before fitting the final screws, since the heads may have been knocked during assembly and need more alignment. Also connections may have pulled apart during assembly. When everying is working perfectly, tighten the case, put away your tools, and return to happy programming.

● *See the Shopping Page for tape head suppliers.*

CHARACTER GENERATOR

Easy guide to generate new character sets

Many people criticise the printing of the Sinclair printer, Mike Biddell has discovered a method to improve the character font to make reading printouts easier

THERE ARE many simple programs to allow one to generate new character sets/fonts for the Spectrum. If you have used one of them you will understand that there is a good deal of work involved in reshaping each letter individually for the lower- and upper-case character sets — 52 characters in all.

Being by nature computer-lazy — if the computer can do it you avoid a good deal of work — I wondered whether the computer could be persuaded to take on the whole task, if given a suitable set of rules. At the outset, I was not sure whether it would be possible but discovered, to my surprise, that two simple sets of transforms would produce a new chunky character set which reduced colour crawl on the VDU and gave more legible, reproducible print from the printer. In addition, given a cassette of pre-recorded software, when loaded it could be converted instantly to the new text with a single poke.

Before determining the rules for transforming the existing character set, it was necessary to examine closely the eight bytes making up each character and the decimal equivalent of those bytes. To do it, I developed the program shown in figure one. When it is entered, pressing any key reveals each letter as a giant 8×8 matrix in the middle of the screen. Some typical screen views are shown in figure two.

I examined all the upper- and lower-case letters and tabulated the decimal values of all the 416 bytes involved. It became apparent immediately that tranforms for the upper-case characters would be detrimental to the lower-case and vice versa. The final program would therefore need to operate on lower- and upper-case separately.

As an example of how a transform is to be achieved, let us examine the capital F — see figure three.

To give the capital F a chunky vertical, we would need to block in the points marked with an X. It should be apparent that a single transform on bytes 3, 5, 6 and 7 — which are all decimal 64, 01000000 binary — incorporated in a program line such as:

10 IF byte = 64 then let byte = 96

would achieve the desired result.

That would produce a binary digit 01100000 in place of the binary digit 01000000 and, in effect, that single program line could double the width of the vertical stem of the F, once the character set is moved to RAM, where you can get at it.

It is something of a headache working-out the set of transforms to produce the desired result. Some which you might think are front-runners work well on the characters at which they are aimed but reduce others to Chinese. So, after many hours of trial and error — mostly error — the first automatic new character generator was born. As an example of the new character font, together with the Sinclair font, see figure four.

The main program is listed in figure five and the following notes will help to complete your understanding of how it works:

Line 10 sets RAMTOP to 63999 to allow room for the character set to be copied out of ROM to 64000 onwards.

Figure 1. A programme to examine the Spectrum character set.

```
  5 LET a=32
  6 LET c=15616
 10 CLS
 15 PRINT AT 0,0;CHR$ a
 20 LET a=a+1
 30 LET y=175
 35 LET x=0
 38 FOR k=y-7 TO y
 40 FOR j=x TO x+7
 50 IF POINT (j,k)=1 THEN PRINT
AT 183-k,j+10; INVERSE 1;" "
 55 IF POINT (j,k)=0 THEN PRINT
AT 183-k,j+10;"."
 60 NEXT j
 62 PRINT AT 23-(183-k),j+10;PE
EK c
 65 LET c=c+1
 70 NEXT k
 80 IF CODE INKEY$>0 THEN GO TO
10
 90 GO TO 80
```

CHARACTER GENERATOR

Lines 40, 50, 60 ensure that the area of RAM to be used is empty to start.

Line 70 points variable 'a' at start of RAM storage.

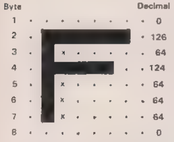
Line 80 defines a 'j loop' for all the addresses of the upper-case letters in the ROM.

Lines 90 to 210 peek the ROM bytes, transform them and store them above RAMTOP.

Lines 215 to 520 repeat that procedure for the lower-case letters, which are stored from addresses 16136 to 16383 in the ROM.

Lines 620 and 630 poke the system variable 'CHARS' — 23606 and 23607, see page 173 of the Sinclair manual — and that points the character generator at 64000 in RAM to read the new set.

Figure 3.



The balance of the program produces a little demonstration — attach your printer — and clears the Basic program, leaving the characters safe above RAMTOP. You can save the new character set as described in line 1080 of the program.

It has therefore been possible to instruct the computer to produce a new set of characters which can be used to good effect in games. In addition, if you require black bold print from your printer, which reproduces well, the program will convert all existing cassette-based software, excluding machine code, for immediate listing from your printer. It is also an interesting exercise which will expand your understanding of character construction using binary notation.

Figure 2. Typical screen views of letters produced by the programme in figure 1.

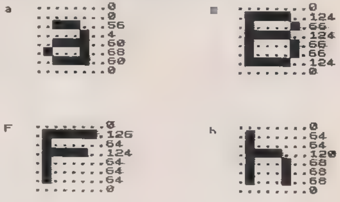


Figure 4. New character font, followed by Sinclair font.

```
INSTRUCTIONS-NEW CHARACTER FONT

1. After this program has self
destructed, poke 23607,248 to
turn on the new character set.

2. Then load your own program in
 the normal way,poke 23607,249
and it is instantly converted
to the new text. Now  use LLIST
to see the enhanced   output
from the zx-printer.

3. Now poke 23607,60 and LLIST.
    Compare this normal listing
with the enhanced one.

4. SAVE 'text' CODE,64000,1536
(this also saves U.D.G's.)


INSTRUCTIONS-NEW CHARACTER FONT

1. After this program has self
destructed, poke 23607,249 to
turn on the new character set.

2. Then load your own program in
 the normal way,poke 23607,249
and it is instantly converted
to the new text. Now  use LLIST
to see the enhanced   output
from the zx-printer.

3. Now poke 23607,60 and LLIST.
    Compare this normal listing
with the enhanced one.

4. SAVE 'text' CODE,64000,1536
(this also saves U.D.G's.)
```

CHARACTER GENERATOR

Figure 5: Main listing.

```
  10 CLEAR 63999
  15 PAPER 0: BORDER 0: CLS
  20 PRINT AT 10,0; INK 7; FLASH
 1; PAPER 1;"PLEASE WAIT-CREATIN
G NEW SCRIPT"
  22 POKE 23607,150
  40 FOR j=64000 TO 65000
  50 POKE j,0
  60 NEXT j
  70 LET a=64000
  80 FOR j=15616 TO 16135
  90 LET b=(PEEK j)
  91 IF b=128 THEN  LET b=b+64
  92 IF b=98 THEN  LET b=102: IF
 b=82 THEN  LET b=114
  95 IF b=70 THEN  LET b=102
 101 IF b=68 THEN  LET b=b+40
 105 IF b=66 THEN  LET b=102
 110 IF b=64 THEN  LET b=b+32
 120 IF b=32 THEN  LET b=b+16
 130 IF b=16 THEN  LET b=b+8
 135 IF b=8 THEN  LET b=b+4
 140 IF b=4 THEN  LET b=b+2
 150 IF b=2 THEN  LET b=b+4
 160 IF b>255 THEN  GO TO 200
 170 IF b=0 THEN  GO TO 200
 180 POKE a,b
 200 LET a=a+1
 210 NEXT j
 215 LET a=64520
 220 FOR j=16136 TO 16383
 222 LET b=PEEK j
 230 IF b=120 THEN  LET b=124
 232 IF b=84 THEN  LET b=214
 233 IF b=104 THEN  LET b=238
 234 IF b=96 THEN  LET b=112
 235 IF b=68 THEN  LET b=102
 238 IF b=64 THEN  LET b=96
 240 IF b=60 THEN  LET b=126
 242 IF b=56 THEN  LET b=60
 243 IF b=48 THEN  LET b=56
 244 IF b=40 THEN  LET b=44
 245 IF b=36 THEN  LET b=38
 248 IF b=34 THEN  LET b=102
 250 IF b=32 THEN  LET b=96
 255 IF b=28 THEN  LET b=60
 260 IF b=24 THEN  LET b=30
 265 IF b=16 THEN  LET b=24
 270 IF b=12 THEN  LET b=30
 280 IF b=4 THEN  LET b=6
 500 POKE a,b
 510 LET a=a+1
 520 NEXT j
 620 POKE 23607,249
 630 POKE 23606,00
 635 PAPER 0
 636 BORDER 0
 640 CLS
 650 PRINT AT 10,0; INK 5; PAPER
 0;"CHARACTERS loaded-© m.P.bidd
ell"
 660 PRINT
 670 PAUSE 200
 680 CLS
 690 GO SUB 1000
 705 PAUSE 100
 800 COPY
 810 POKE 23607,60
 815 CLS
 820 GO SUB 1000
 840 COPY
 850 PAUSE 100
 860 NEW
1000 PRINT  INK 6;"INSTRUCTIONS-
NEW CHARACTER FONT"
1010 GO SUB 2000
1020 PRINT  INK 5;"1. After this
 Program has self  destructed, P
oke 23607,249 to   turn on the n
ew character set."
1030 GO SUB 2000
1040 PRINT  INK 5;"2. Then load
your own Program in the normal w
ay,Poke 23607,249  and it is ins
tantly converted   to the new te
xt. Now  use LLIST to see the en
hanced   output    from the zx-P
rinter."
1050 GO SUB 2000
1060 PRINT  INK 5;"3. Now Poke 2
3607,60 and LLIST.      Compare t
his normal listing with the enha
nced one."
1070 GO SUB 2000
1080 PRINT  INK 5;"4. SAVE 'text
' CODE,64000,1536  (this also sa
ves U.D.G's.)"
1090 GO SUB 2000
1100 RETURN
2000 PRINT : FOR j=20 TO 30
2010 BEEP .1,-j
2020 NEXT j
2030 RETURN
```

SOUND BOARD

Making your ZX-81 into a big noise

Last issue we produced a sound generation board for the Spectrum. Graham Bradley adapts that for the older of Sinclair's machines

ONE OF the shortcomings of the ZX-81 is that it has no facilities for sound generation. You may be compelled by financial restraints to be content with black-and-white low-resolution graphics — just consider the cost of a colour TV — but the addition of a sophisticated sound output may cost less than you think and this add-on uses only four integrated circuits.

In addition to three channels of sound with tone, duration and envelope control independent of processor operation, there are two input/output ports available. If that is not enough you can also use it with a Spectrum when you can afford a colour TV.

The full circuit diagram is shown in figure one and the Veroboard layout is in figure two. If the sound generator is for use solely with the ZX-81, the clock circuit may be omitted and pin 22 of the PSG can be connected to the clock output of the ZX-81 — pin 6 bottom on the edge connector. See edge connector signal allocation on page 47.

Table one from the original article is reproduced to assist you with programming the device. The first six registers are grouped in pairs. Each pair is used to provide a 12-bit word which controls the tone period or the pitch of the channel sound. A number between 0 and 15 placed in R1 will produce a big change in the output pitch or frequency of channel A.

A number greater than 15 will have no additional effect because only the bottom four bits of registers R1, R3 and R5 are used. The fine control of pitch for channels A, B and C is provided by the 8-bit binary number

Program 1.

```
  1 REM 0123456789012
 10 REM ZX-81 M/C LOADER
 20 PRINT "ENTER HEX DATA ■ DIG
ITS AT A     TIME"
 30 FOR A=16514 TO 16526
 40 PRINT A;"     ";
 52 INPUT H$
 60 PRINT H$
 70 LET D=16*CODE H$(1)+CODE H$
(2 TO )-476
 80 PRINT D
 90 POKE A,D
100 NEXT A
```

Table 1.

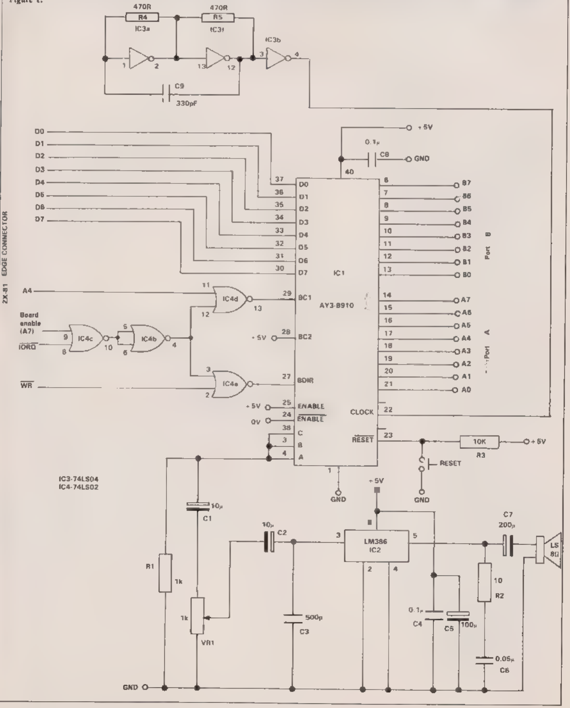
ENTER HEX DATA 2 DIGITS AT A TIME

16514	11	17
16515	00	0
16516	00	0
16517	01	1
16518	6F	111
16519	FF	255
16520	ED	237
16521	59	89
16522	0E	14
16523	7F	127
16524	ED	237
16525	51	81
16526	C9	201

Program 1A.

```
  1 REM )  ■? COPY GOSUB ?:■ GO
SUB ?TAN
 10 REM ZX-81 M/C LOADER
 20 PRINT "ENTER HEX DATA 2 DIG
ITS AT A     TIME"
 30 FOR A=16514 TO 16526
 40 PRINT A;"     ";
 50 INPUT H$
 60 PRINT H$,
 70 LET D=16*CODE H$(1)+CODE H$
(2 TO )-476
 80 PRINT D
 90 POKE ■,D
100 NEXT A
```

Figure 1.


470R
R4
IC3a
470R
R5
IC3f
IC3b
C9
330pF
+5V
0.1µ
C8
GND
D0
D1
D2
D3
D4
D5
D6
D7
ZX-81 EDGE CONNECTOR
A4
Board enable (A7)
IORQ
IC4c
IC4b
IC4d
IC4a
WR
+5V
0V
IC1
AY3-8910
BC1
BC2
BDIR
ENABLE
CLOCK
RESET
B7
B6
B5
B4
B3
B2
B1
B0
Port B
A7
A6
A5
A4
A3
A2
A1
A0
Port A
10K
R3
RESET
GND
IC3-74LS04
IC4-74LS02
10µ
C1
R1
1k
1k
VR1
10µ
C2
500p
C3
LM386
IC2
C7
200µ
LS
8Ω
10
R2
0.1µ
C4
C5
100µ
0.05µ
C6
GND

placed into the R0, R2 and R4 registers respectively. Thus a number between 0 and 255 placed in register R0 will enable you to make fine adjustments to the pitch of channel A. Each of the 15 coarse pitches selected by R1 will have 255 fine variations selected by R0.

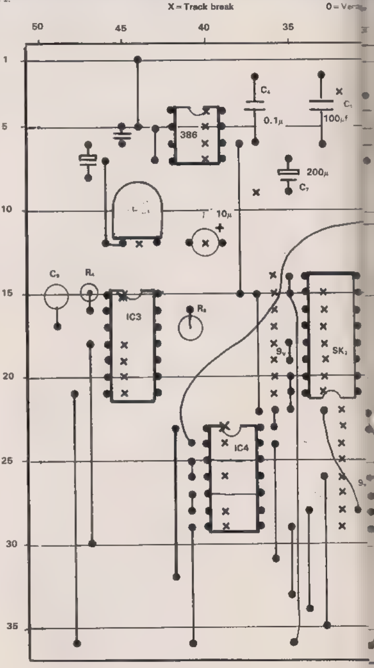
The amplitude of the tone for channels A, B and C can be controlled by the lower four bits of registers, R8, R9, R10 — again a number between 0 and 15. If the fifth bit is set to one by placing 24 in registers R8, R9 or R10 the output of the channel will be affected by the envelope generator. The various envelope shapes available are shown in figure three from the original article and are obtained by placing a number between 0 and 15 in R13.

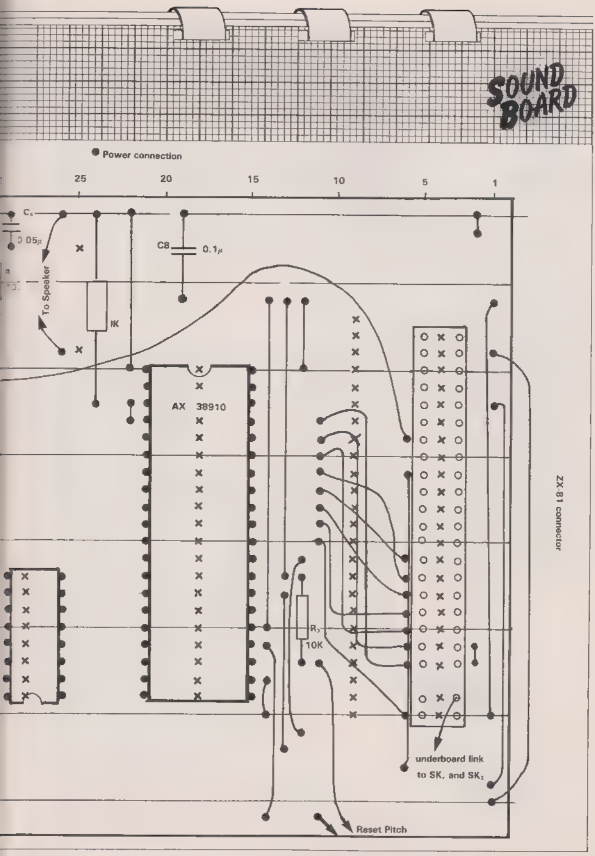
Register R7 is very important as it is used to enable the tone generator or the noise generator on the appropriate channel. It is also used to determine whether the I/0 ports are used for input or output. Putting a zero note, not a one, into the appropriate

Program 4.

```
  1 REM "PHOTON CANNON"
 10 LET Z=1
 20 FOR A=0 TO 15
 30 POKE 16515,1
 40 POKE 16516,A
 50 RAND USR 16514
 60 POKE 16515,0
 70 POKE 16516,5
 80 RAND USR 16514
 90 POKE 16515,7
100 POKE 16516,56
110 RAND USR 16514
120 POKE 16515,8
130 POKE 16516,24
140 RAND USR 16514
150 POKE 16515,12
160 POKE 16516,6
170 RAND USR 16514
180 POKE 16515,13
190 POKE 16516,0
200 RAND USR 16514
210 NEXT A
220 LET Z=Z+1
230 IF Z=4 THEN STOP
240 GOTO 20
```

Figure 2.




SOUND BOARD
Power connection
25
20
15
10
5
1
C8
0.1μ
0.05μ
To Speaker
1K
AX 38910
R
10K
underboard link to SK, and SK₂
Reset Pitch
ZX-81 connector

SOUND BOARD

bit will enable the tone or noise generators. Thus 7 — binary 00000111 — is used to turn off the tone generator and turn on the noise generator. A decimal value of 56 — binary 00111000 — will turn on the tone generators and turn off the noise generator.

The repetition rate of the envelope is determined by the numbers placed in registers R11 and R12. A number from 0 to 255 in R12 is the coarse tune for the repetition rate.

The example listed will give some idea of the use of the registers. If you write programs which are of interest to other readers we will be prepared to publish them in future issues.

The board will work with both the Spectrum and the ZX-81. A disadvantage of the ZX-81 is that it has no Basic IN or OUT commands. We shall overcome that omission with a machine code routine held in the usual place in a REM statement at the start of the program.

Program one can be used to load the hexadecimal data listed in table one into the REM line. The appearance of the program after the data has been entered will change to that of program 1a. Table one also lists the hex and decimal data if you prefer to use a decimal loader. Comparison of the decimal data with the codes listed at the back of the Sinclair ZX manual will explain the appearance of the REM statement in program 1a.

Program 5.

```
   1 REM "NOTE MAKER"
  10 PRINT AT 20,0;"PITCH(0-15)"
  20 INPUT A
  30 PRINT AT 21,0;"DURATION"
  40 INPUT B
  50 POKE 16515,1
  60 POKE 16516,A
  70 RAND USR 16514
  80 POKE 16515,8
  90 POKE 16516,15
 100 RAND USR 16514
 110 POKE 16515,7
 120 POKE 16516,56
 130 RAND USR 16514
 140 POKE 16515,0
 150 POKE 16516,16
 160 RAND USR 16514
 170 POKE 16515,12
 180 POKE 16516,13
 190 RAND USR 16514
 200 POKE 16515,13
 210 POKE 16516,0
 220 RAND USR 16514
 230 CLS
 240 GOTO 10
```

Program 2.
Subroutine.

```
9000 REM SUB TO SET A REGISTER
9010 POKE 16515,REGADDRESS
9020 POKE 16516,REGDATA
9030 RAND USR 16514
9040 RETURN
```

Program 3.
Machine code.

```
4082 11 00 00     LD DE,0000
4085 01 BF FF     LD BC,FFBF
4088 ED 59        OUT (C),E
408A 0E 7F        LD C,7F
408C ED 51        OUT (C),D
408E C9           RET
```

ARTICLE OUTLINE

The good author's guide to explaining projects

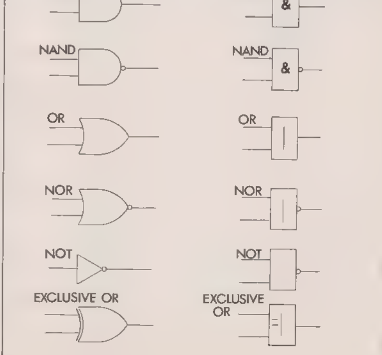
IF YOU WISH to submit articles to *Sinclair Projects* we would appreciate it if you adhere to the following rules. Although they are not exclusive it would help us to evaluate projects if there is some element of compatibility between different presentations.

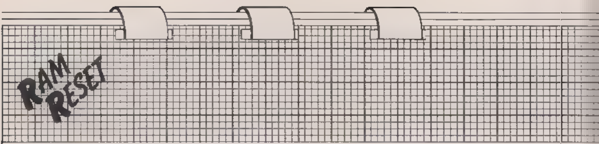
It would also make it much easier for us to publish the articles without errors as there would be less chance of confusion about meanings. The main points to note are:

- All manuscripts should be typed with double-line spacing.
- Logic symbols should follow international standards.
- Circuit symbols should follow international standards.
- Circuit diagrams should have the values of the components shown, not a reference to a component table.
- Parts of integrated circuits should be designated with a note on the diagrams—IC5 - 74LS14, for example.
- All circuits should be designed for construction using standard Veroboard. Any printed circuit board designs are likely to be returned for conversion. Submission of a project on a PCB will not exclude future publication.
- Any constructional detail which is unusual or slightly complicated should be illustrated with simple hand-drawn diagrams, showing how it can be implemented.

For those who are familiar with British Standards logic symbols, they are shown here, along with the international symbols.

- Where projects are designed to plug into the rear of the computer they should be built on the 36 strips × 50 holes size of Veroboard with the board vertical and an extender card at the rear to allow other projects to be stacked. For Spectrum projects the connector should be central on the board with four strips spare at each side and one row of holes spare beneath the connector. For ZX-81 projects the connector should have two rows of holes spare beneath it with seven spare strips at the right-hand side. Where that is impracticable, boards may be remote and connected by ribbon cable to a socket and extender card assembly.
- Components should be available to hobbyists through the normal retail channels and where a component is not a widely-stocked item, sources should be given.





Running automatically at the press of a switch

Following the battery-backed RAM project in our last issue Graham Bradley has added a small circuit which provides two useful facilities, autorun and program interrupt

THIS SMALL circuit when used in conjunction with the battery-backed RAM board or a single EPROM provides two useful facilities. It enables programs in the CMOS RAM to be run by a single depression of a switch. That is the autostart feature. It also enables a program to be interrupted, i.e., re-set, without destroying the contents of RAM. That is particularly useful when writing and debugging programs in machine code or Basic.

A low signal on the Z-80 RESET line initialises the CPU. The CPU initialisation includes: set the program counter to zero; disable the interrupt (INT); sets register I = 00; sets register R = 00; sets interrupt mode 0.

During the re-set time the address and data bus go to a high-impedance state, all the control signals to the inactive state, and no refresh occurs.

After a re-set the first operation of the CPU is to perform an opcode fetch from location 0000. The program in the ZX-81 ROM beginning at 0000 sets various initial states and calls a routine which checks and clears all memory from 16K upwards. Because A15 of the battery-backed RAM is not decoded it will also clear that RAM. Thus it will effectively wipe out any program in the normal RAM space.

Figure 7. How A13 is forced to go high.

A8	Q2	A13	A13
1	0	0	1
0	0	1	1
1	1	0	1
0	1	1	0

NAND

The circuit shown here causes A13 to be pulled high until the second M1 cycle is received after a re-set. Instead of fetching an opcode from 0000 the opcode will come from 2000 — 8192 or 8K. That is the first location in the battery-backed RAM. The instruction could be a jump to 2004, so that the CPU performs the routine at that location in RAM immediately after the re-set signal, hence avoiding the memory clear routine.

A Basic program will crash unless certain variables are initialised, so you will need a thorough understanding of the ROM to use this feature with a Basic program. It will prove useful, however, for re-starting ma-

Figure 5. Switch panel.

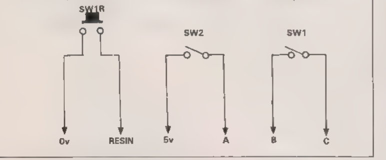


Figure 1. Circuit diagram of normal ZX-81 re-set button.

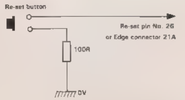
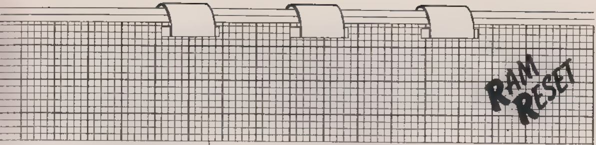


Table 1. Re-set programs.

Decimal Address	Decimal Data	Hex Address	Hex Data	Comments
		Usual re-set		
8192	195	2000	C3	JMP to
8193	0	2001	00	0000
8194	0	2002	00	
or				
8192	199	2000	C7	RST 0
		Warm re-set		
8192	195	2000	C3	Jump to
8193	04	2001	03	8195d
8194	32	2002	20	2003h
8195	(warm re-set)	2003	(warm re-set routine)	



chine code routines. The time constant of the power-on-re-set capacitor/resistor combination is too long to work with this circuit. The 1uF capacitor C5 which is near the CPU on the ZX-81 PCB should be replaced by a 47nF ceramic capacitor.

The computer should still re-set correctly, if not increase the value of C5 slightly. IC 8a and IC 8b are configured as two monostables. The input to IC 8a is pulled high by R2 so that D2 is usually low. After two M1 pulses Q2 will be low and Q1 will be high and they are normally in this state A13' = A13.

When SW1 is closed D2 goes high for a short time. At the next M1 cycle Q2 will go high and the output of IC 8b will go low causing a RESET. SW2 must be closed to reduce the re-set time constant. It is opened to allow power-on-re-set. M1 pulses are suspended during a re-set but the delay in discharging C5 on the ZX board means that the high on D1 will be clocked through to Q1 before the M1 pulses cease. Thus $\overline{Q1}$ will be low, causing A13 to be high.

When the re-set line goes high again the D2 input will be low and the Q2 output will be high and Q1 will be low. The first M1 pulse will clock the

Figure 8. Inverting A13 switches from 0000H ≡ 2000H.

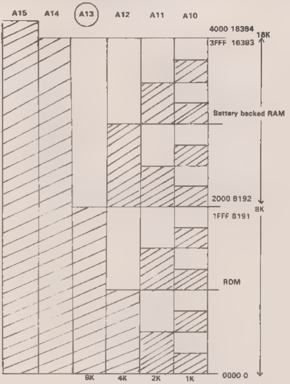
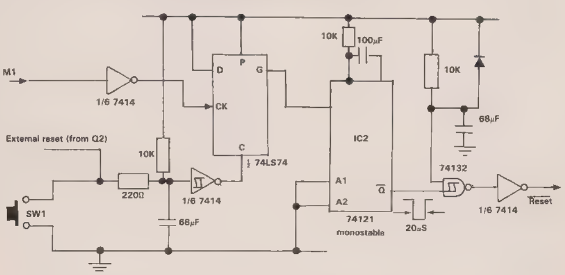


Figure 2. Manual & power-on reset, circuit recommended by Mostek for use with dynamic RAM



low on D2 to Q2 but Q1 will still be low so that the first opcode fetch will be from 2000H not 0000H — 8192D not 0D. The second M1 pulse will cause $\overline{Q1}$ to go high, so releasing A13 so that all subsequent addresses are normal.

There is just sufficient room on the battery RAM board to add this circuit, or it can be built on a separate piece of Veroboard. The switches will need to be mounted in a suitable place as there is no room for them on the RAM/PROM board.

The requirements for the RESET pulse when dealing with dynamic RAM are very stringent. The RESET pulse must be synchronised with M1, which in this case it is, so that a memory access is not aborted in the middle of its cycle causing a short access of the dynamic RAM and hence the destruction of data in the RAM. The duration of the RESET pulse must be less than 20µS to avoid suspending the CPU refresh of dynamic RAM for a sufficient time to destroy data in the RAM. The circuit recommended by the manufacturers for a manual and power-on-re-set is shown in figure two.

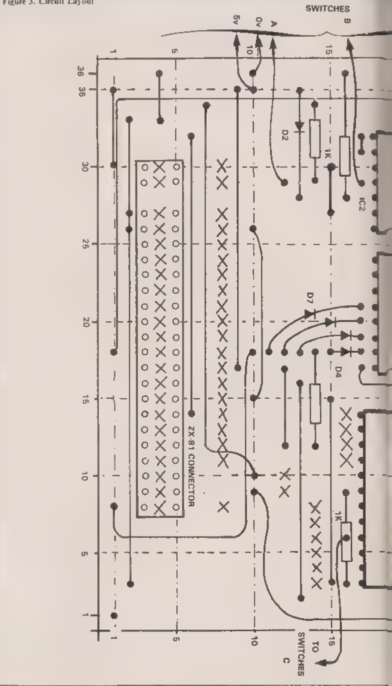
IC 2 is a monostable which gives a much more predictable output period than those in the circuit we have used. If you decide to add the components to the RAM board, first re-route the output from Y7 (pin 7) of the 74LS138 (IC1). Take it close to RAM 2 and then to pin 18 of RAM 4 socket.

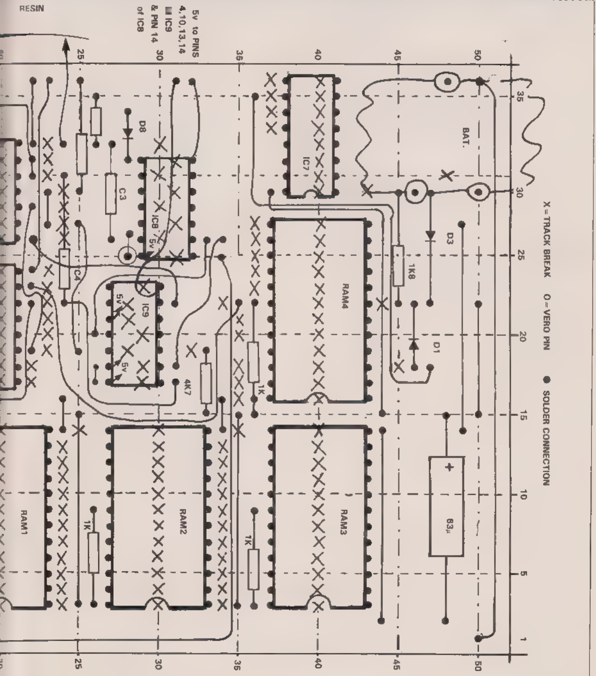
Mount the IC sockets as shown on the layout and solder in the components and links. Underboard links are made from the 5V rail to pins 1,4,10,13, and 14 of IC9 and to pin 14 of IC8. Links are also made from 0V to pin 7 of both ICs 8 and 9.

The write enable from pin 11 of IC2 is taken to the write protect switch (B) and from there to pin 21 of RAM 1 (C). One pushbutton and two toggle switches are required. They are mounted on a separate board which rests on top of the ZX-81 and is connected to the RAM board by six wires labelled 0V, 5V, RESIN, A, B, C on the diagrams — figure five.

Remove the link from pin 1 to pin

Figure 3. Circuit Layout




RESIN
5v to PINS 4,10,13,14 ■ IC9 & PIN 14 of IC8
X = TRACK BREAK O = VERO PIN ● SOLDER CONNECTION
BAT.
IC7
IC8
IC9
RAM1
RAM2
RAM3
RAM4
D1
D3
D8
C3
C4
1K8
4K7
1K
63µ
5v

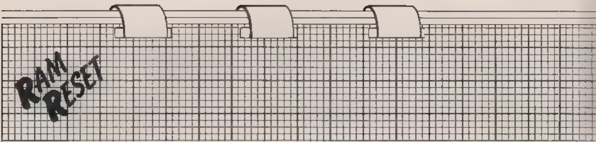
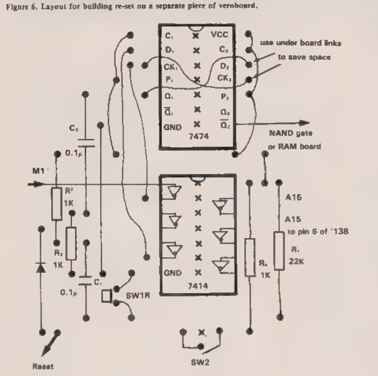


Figure 6. Layout for building re-set on a separate piece of veroboard.



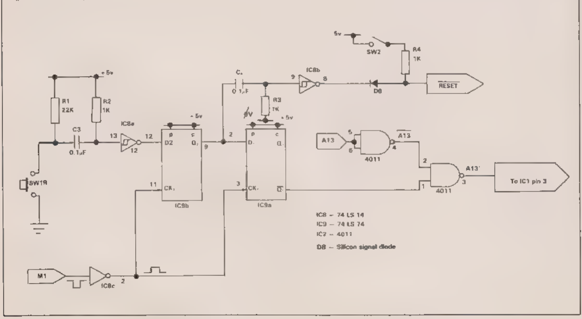
14 of IC2 on the RAM board and fit the link from IC9 pin 6 to IC2 pin 1.

To test the operation of the circuit, load the normal re-set program which causes the computer to jump to 2000H and then back to 0000H. The computer should re-set as usual and clear the memory. Then write a normal re-set routine, including instructions to load the I register with 1E. The CPU will re-set but will not clear the memory.

A second pushbutton can be connected so that it takes the re-set line to ground through a 100ohm resistor. It can be used instead of pulling-out the power plug to provide a normal ZX-81 reset.

The feature has already been fitted to the author's ZX-81 and so is not shown on this add-on unit.

Figure 4. Re-set circuit diagram.



EDGE CONNECTOR

Edge Connector signal allocation

SPECTRUM

BOTTOM (SIDE B)	Pin	TOP (SIDE A)
A_{14}	1	A_{15}
A_{12}	2	A_{13}
5v	3	D_7
9v		
		Slot
0v		D_0
0v	7	D_1
CK	8	D_2
A_0	9	
A_1	10	D_5
A_2	11	D_3
A_3	12	D_4
$\overline{IORQGE}$	13	$\overline{INT}$
0v		$\overline{NMI}$
Video	15	$\overline{HALT}$
Y	16	$\overline{MEMRQ}$
V	17	$\overline{IOREQ}$
U	18	$\overline{RD}$
$\overline{BUSRQ}$	19	$\overline{WR}$
$\overline{RESET}$	20	−5v
A_7	21	$\overline{WAIT}$
A_6	22	+12v
A_5	23	−12v
A_4	24	$\overline{MI}$
$\overline{ROMCS}$	25	$\overline{RFSH}$
$\overline{BUSAK}$	26	A_8
A_9	27	A_{10}
A_{11}	28	

ZX-81

BOTTOM (SIDE B)	Pin	TOP (SIDE A)
5v	1	D_7
9v	2	$\overline{RAMCS}$
	3	Slot
0v	4	D_0
0v	5	D_1
$\varnothing$	6	D_2
A_0	7	D_6
A_1	8	D_5
A_2	9	D_3
A_3	10	D_4
A_{15}	11	$\overline{INT}$
A_{14}	12	$\overline{NMI}$
A_{13}	13	$\overline{HALT}$
A_{12}	14	$\overline{MREQ}$
A_{11}	15	$\overline{IORQ}$
A_{10}	16	$\overline{RD}$
A_9	17	$\overline{WR}$
A_8	18	$\overline{BUSAK}$
A_7	19	$\overline{WAIT}$
A_6	20	$\overline{BUSRQ}$
A_5	21	$\overline{RESET}$
A_4	22	$\overline{MI}$
$\overline{ROMCS}$	23	$\overline{RFSH}$

GET THE RIGHT ANGLE ON YOUR SPECTRUM

Trickstick turns your Spectrum into the most sophisticated games machine in the world. Your fingers rest on the Trickstick's six sensitive pads (four directions and two fire buttons) and the harder you press the faster you go or the harder you turn.

Trickstick works by picking up mains hum from your body and converting it by an ingenious circuit design directly into digital input. Proportionality gives vast possibilities for more interesting games.

- Each Trickstick comes with its own interface included in the price.
- Up to eight interfaces can be stacked into each other for 1-8 simultaneous players. (Own 1-8 player game is Attaktics at £10).
- Works with most Kempston compatible software, and with our new programmable adaptor (£10 for Trickstick owners) it works with ANY software.
- Easy to program, even for proportional games.
- In the shops soon. Now available by mail telephone order at £34.50.

TRICKSTICK-
The revolution that Runs Rings round ordinary joysticks

MAIL ORDER ADVERTISING
British Code of Advertising Practice

Advertisements in this publication are required to conform to the British Code of Advertising Practice. In respect of mail order advertisements where money is paid in advance, the code requires advertisers to fulfil orders within 28 days, unless a longer delivery period is stated. Where goods are returned undamaged within seven days, the purchaser's money must be refunded. Please retain proof of postage/despatch, as this may be needed.

Mail Order Protection Scheme

If you order goods from Mail Order advertisements in this magazine and pay by post in advance of delivery, *Sinclair User* will consider you for compensation if the Advertiser should become insolvent or bankrupt, provided:

(1) You have not received the goods or had your money returned; and

(2) You write to the Publisher of *Sinclair User* summarizing the situation not earlier than 28 days from the day you sent your order and not later than two months from that day.

Please do not wait until the last moment to inform us. When you do write, we will tell you how to make your claim and what evidence of payment is required.

We guarantee to meet claims from readers in accordance with the above procedure as soon as possible after the Advertiser has been declared bankrupt or insolvent (up to a limit of £4,250 per annum for any one Advertiser so affected and up to £10,000 per annum in respect of all insolvent Advertisers. Claims may be paid for higher amounts, or when the above procedure has not been complied with at the discretion of *Sinclair User*, but we do not guarantee to do so in view of the need to set some limit to this commitment and to learn quickly of readers' difficulties).

This guarantee covers only advance payment sent in direct response to an advertisement in this magazine (not, for example, payment made in response to catalogues etc, received as a result of answering such advertisements). Classified advertisements are excluded.

Upgrade your 16K ZX SPECTRUM Now!

The CHEETAH 32K RAMPACK simply plugs into the user port at the rear of your computer and increases the memory instantly to 48K.

- ★ Fully compatible with all accessories via rear edge connector
- ★ No need to open computer and invalidate guarantee
- ★ Why send your computer away and wait weeks for upgrade
- ★ Fully cased tested and guaranteed.

Why wait any longer?

Only £39.95 including VAT and P&P.

Now make your Spectrum and ZX-81 Talk

The Cheetah "SWEET TALKER" just plugs into the back of the computer using the existing power supply. Based on an allophone system you can easily program any word sentence or phrase. Fully cased, tested guaranteed and compatible with all accessories via rear edge connector. Complete with demonstration cassette and full instructions. No more lonely nights! Simply incredible at **£29.75** (Please quote when ordering whether Spectrum or ZX81 owner)

16K RAM Pack for ZX-81	£19.75
64K RAM Pack for ZX-81	£44.75

Prices include VAT, postage & packing. Delivery normally 14 days. Export orders at no extra cost. Dealer enquiries welcome.

Send cheque/PO now to:
CHEETAH MARKETING LTD
Dept SPJ
24 Ray Street
London EC1 R3 DJ
Tel: 01-278 6954

32K RAM Pack and "SWEET TALKER" also available from larger Branches of

John Menzies Boots WH SMITH